CORRELATION-BASED COMMUNICATION IN WIRELESS MULTIMEDIA SENSOR NETWORKS

A Thesis
Presented to
The Academic Faculty

by

Rui Dai

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

CORRELATION-BASED COMMUNICATION IN WIRELESS MULTIMEDIA SENSOR NETWORKS

Approved by:

Professor Ian F. Akyildiz, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chuanyi Ji
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Mostafa Ammar
College of Computing
Georgia Institute of Technology

Professor Ye Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: August 16, 2011

To my family,

for their endless love and support.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Professor Ian F. Akyildiz, for his invaluable guidance and support during my PhD study. Professor Akyildiz has been a wonderful advisor. I am always inspired by his breadth of knowledge, approaches to solve problems, and vision on new research problems. I have benefited a lot from his professional training on how to conduct top-notch research. I have also been fortunate to listen to his stories on life experiences, which will definitely be helpful in my future life and career.

A special thank goes to Drs. Chuanyi Ji, Ye Li, Justin Romberg, and Mostafa Ammar, who kindly agreed to serve in my dissertation defense committee. Their valuable comments and suggestions have helped to improve the quality of this dissertation.

I am especially grateful to my friend and colleague, Pu Wang, for all the valuable work done together that leads to the completion of this dissertation. Furthermore, I want to thank all former and current members of the Broadband Wireless Networking (BWN) Laboratory for their assistance in research as well as friendship. It has been a pleasure to work with all the talented people in the BWN family.

I would like to thank all my family members in China. They bless for me all the time and give me constant support and encouragement. I am very grateful to my husband Junjie Zhang. He has helped me greatly during the past four years. Without his love and support, I would not have been able to complete this work.

Last but not least, I would like to thank the many anonymous reviewers that with their comments greatly improved the content of the papers from which this dissertation has been partly extracted.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Wireless multimedia sensor networks (WMSNs) are networks of wirelessly interconnected devices that are able to retrieve multimedia content such as video and audio streams, still images, and scalar data from the environment. Most applications of WMSNs require the delivery of multimedia information with a certain level of quality of service (QoS). This is a challenging task because sensors are constrained in battery and processing capabilities, while the delivery of multimedia flows is a resource-intensive task. In a densely deployed sensor network, there exists correlation among the observations of camera sensors with overlapped coverage areas, which could be exploited to remove data redundancy in the network.

The objective of this thesis is to design efficient communication protocols for WMSNs by leveraging the correlation of visual information of camera sensors. First, the spatial correlation of visual information in WMSNs is studied. By studying the sensing model and deployments of cameras in a WMSN, a spatial correlation coefficient is derived to describe the degree of correlation of visual information observed by cameras with overlapped field of views. The joint effect of multiple correlated cameras is also studied. An entropy-based analytical framework is developed to measure the amount of visual information provided by multiple correlated cameras.

The compression performance of correlated visual information is then studied. A collaborative clustered compression framework is proposed with an objective to maximize the overall compression gain of the visual information gathered in WMSNs. To achieve this, an Entropy-based Divergence Measure (EDM) scheme is proposed to predict the compression efficiency of performing joint coding on the images collected by spatially correlated cameras. Utilizing the predicted results from EDM,

a Distributed Multi-cluster Coding Protocol (DMCP) is proposed to construct a compression-oriented coding hierarchy.

The correlation of visual information is then utilized to design a network scheduling scheme to maximize the lifetime of WMSNs. The scheduling scheme consists of three components including MinMax Degree Hub Location (MDHL), Minimum Sum-entropy Camera Assignment (MSCA), and Maximum Lifetime Scheduling (MLS). The MDHL problem finds the optimal locations to place the multimedia processing hubs such that the number of channels required for frequency reuse is minimized. The MSCA problem assigns each camera sensor to a hub in such a way that the global compression gain is maximized by jointly encoding the correlated images gathered by each hub. At last, the MLS problem finds a schedule for the cameras to maximize the network lifetime by letting highly correlated cameras perform differential coding.

Furthermore, a correlation-aware QoS routing algorithm is designed to efficiently deliver visual information under QoS constraints. A correlation-aware inter-node differential coding scheme is introduced to remove traffic redundancy along routing paths, and a correlation-aware load balancing scheme is proposed to prevent network congestion by splitting the correlated flows that cannot be reduced to different paths. These correlation-aware operations are integrated into an optimization QoS routing framework that minimizes energy consumption subject to delay and reliability constraints.

CHAPTER I

INTRODUCTION

1.1 Background

The integration of low-power wireless networking technologies with inexpensive hardware such as complementary metal-oxide semiconductor (CMOS) cameras and microphones is now enabling the development of distributed, networked systems that we refer to as wireless multimedia sensor networks (WMSNs) [5]. In WMSNs, interconnected sensor devices collaborate to retrieve multimedia information such as video and audio streams, still images, and scalar data from the environment. Apart from the primary goal of retrieving multimedia data, WMSNs will also be able to store, process in real time, correlate and fuse data originated from heterogeneous sources. These networks can be an integral part of systems such as security surveillance, traffic enforcement, health care delivery, environmental monitoring, and industrial process control.

There are several factors that mainly influence the design of a WMSN [5]. First of all, most WMSN applications require the delivery of multimedia information with a certain level of quality of service (QoS), in terms of delay, jitter, bandwidth, reliability, etc. To achieve this, one major factor we need to consider is the resource constraint: sensor devices are constrained in terms of battery, memory, processing capability, and achievable data rate. In multi-hop wireless networks, the capacity and the delay attainable on each wireless link are location dependent, vary continuously, and may be bursty in nature, thus making QoS provisioning a challenging task. There is also interdependence among functions handled at all layers of the communication stack. This interdependence must be explicitly considered when designing communication

protocols aimed at QoS provisioning. In addition, multimedia contents, especially the visual information (still images and video streams), require high bandwidth to be delivered. Therefore, efficient multimedia processing and compression techniques must be leveraged. Furthermore, the processing and delivery of multimedia contents are not independent, and their interaction has a major impact on the achievable QoS. It is necessary to investigate how to jointly optimize the performances of processing and delivery of multimedia contents.

Protocols for providing QoS support in WMSNs have been proposed at different layers of the communication stack. In the physical layer, the ultra wide band (UWB) technology is proposed since it enables low power consumption and high data-rate communication on simple-design and low-cost radios [44]. MAC layer protocols have been designed that minimize energy consumption and provide application-specific QoS by differentiating network services based on priority levels [54, 38]. In the network layer, various routing protocols have been proposed to provide probabilistic QoS guarantees by regulating traffic to proper paths in the network [20, 29, 11, 10]. For example, the MMSPEED protocol [20] provides probabilistic guarantee in delay and reliability through non-deterministic geographic and multi-path forwarding. As for the transport layer, protocols such as Reliable Multi-Segment Transport (RMST) [59] or the Pump Slowly Fetch Quickly (PSFQ) [63] can be used to enhance reliability as they buffer packets at intermediate nodes, allowing for faster retransmission in case of packet loss. A cross-layer QoS communication module is also designed in [45] to provide an integrated solution for the communication in WMSNs.

To achieve efficient delivery of multimedia contents in a WMSN, apart from properly regulating the traffic flows from the networking perspective as introduced above, it is necessary to reduce the redundancy in the network through various application layer-level technologies [46]. Collaborative multimedia in-network processing is suggested as an effective way to reduce the redundancy in network traffic, where sensor

nodes can filter out uninteresting events locally, or coordinate with each other to aggregate correlated data [5]. In [60] and [28], application-independent task mapping and scheduling solutions are designed to schedule computation tasks and associated communication events for WMSNs. In particular, many schemes have been proposed on the compression and transmission of visual information (images and video streams) in WMSNs. Low complexity image compression schemes are proposed in [39], and algorithms aiming to efficiently transmit images in sensor networks are proposed in [75] and [67].

In WMSNs, multiple camera sensors are deployed to provide multiple views, multiple resolutions, and enhanced observations of the environment [14]. There exists correlation among the visual information observed by cameras with overlapped field of views. This causes substantial redundancy in the network traffic. Previous work on the correlation in sensor networks, such as the theoretical spatio-temporal correlation model in [61], is designed for scalar data applications. As visual information is much more complex than scalar data, the model in [61] cannot be directly applied to visual information.

On the other hand, from the field of multimedia processing, image processing techniques have been used to obtain correlation and design collaborative processing. In [62], images from correlated views are roughly registered using correspondence analysis. Each sensor transmits a low-resolution version of a common area, and the sink combines multiple low-resolution versions into a high-resolution image. In [71], spatial correlation is obtained by an image shape matching algorithm, while temporal correlation is calculated via background subtraction. Based on the spatial and temporal correlation, images from correlated sensors are transmitted collaboratively. However, the performance of image processing algorithms are application dependent: different types of images will require different processing schemes [25]. Moreover, since image processing techniques are complicated, it will bring about extra computation costs

for sensor nodes.

To this end, it is necessary to find light-weighted approaches to obtain the correlation of visual information in WMSNs. Based on the correlation of visual information, we can develop schemes to remove the redundancy in network traffic and also exploit correlation to design efficient QoS communication protocols.

1.2 Research Objectives and Solutions

WMSNs allow the retrieving of video and audio streams, still images, and scalar data from the environment. Of the different types of information in WMSNs, visual information (still images and video streams) is the dominating part of traffic. Visual information requires sophisticated techniques to process and high bandwidth to deliver, which poses a major challenge for the design of WMSNs. It has been found that the observation from camera sensors with overlapped field of views (FoVs) are correlated with each other. By exploiting the correlation of camera sensors, we can remove the redundancy of network traffic and achieve efficient delivery of visual information in WMSNs.

The objective of this thesis is to develop a correlation-based communication framework by exploiting the correlation of visual information observed by camera sensors. First, we study the correlation characteristics of visual information in WMSNs and develop a spatial correlation model that estimates the degree of correlation between camera sensors with overlapped FoVs. Based on the correlation model, we then design a collaborative image compression framework to reduce the redundancy of correlated camera sensors. We also make use of correlation in the network scheduling process and propose a correlation-based scheduling algorithm to maximize network lifetime. To provide better QoS support for WMSN applications, we develop a correlation-aware QoS routing scheme that provides probabilistic QoS support for the delivery of visual information while minimizing energy consumption.

1.2.1 A Spatial Correlation Model for Visual Information in WMSNs

As our first step, we study the correlation characteristics of visual information in WMSNs. In many recent studies on WMSNs, image processing techniques, such as shape matching and image registration, are used to obtain correlation and to design collaborative processing schemes [62, 71]. However, there are some drawbacks to implement image processing based methods in WMSNs [15]. On the one hand, image processing methods are very complicated and can result in considerable energy consumption for sensors. On the other hand, since the performance of image processing methods is usually dependent on the specific features of images, we usually need to choose different image processing methods based on specific applications and image types, and thus, it is necessary to find out a more general way to estimate correlation.

Considering the above factors, we avoid using specific image processing methods; in contrast, we develop a novel analytical spatial correlation coefficient based on the projection geometry of camera sensors [15]. This correlation coefficient just takes in a few parameters such as cameras' locations, sensing directions, and focal lengths as input, and calculates the degree of correlation for any two cameras with overlapped field of views. This correlation coefficient allows us to estimate the degree of correlation for camera sensors through very low communication and computation costs.

Furthermore, we propose an entropy-based framework that addresses how much information can be gained from multiple correlated cameras. By definition, the joint entropy of multiple cameras is calculated from the joint probability distribution of multiple images [12]. To estimate the joint probability distribution is challenging because of the difficulty in image modeling and the high computation and communication costs involved. Instead of estimating the joint probability distribution, we estimate the joint entropy as a function of the spatial correlation coefficients between camera sensors.

The spatial correlation coefficient and the entropy-based framework together complete the derivation of the correlation model. As a simple application of this model, we study a camera selection problem: when multiple correlated cameras can observe the same area of interest, how to select the minimum number of cameras that provides the maximum amount of information for an application. This problem is equivalent to maximizing the joint entropy for a certain number of cameras, and it is solved using our entropy-based framework.

1.2.2 A Collaborative Image Compression Framework Using Clustered Source Coding

The spatial correlation of camera sensors causes substantial redundancy in the observed visual information in WMSNs, which can be removed to improve energy efficiency and network performance. In this work, we design a collaborative image compression framework that maximizes the overall compression gain of WMSNs [65]. We propose to remove the redundancy of visual information through joint compression/coding among multiple correlated cameras. To maximize the compression gain of the whole network, we partition the network into a set of clusters where camera sensors with high joint compression gains are grouped together.

Since entropy serves as the lower bound of coding rates, we can estimate the joint entropy of multiple cameras as a prediction of the joint coding efficiency. We refer to this estimation as Entropy-based Divergence Measure (EDM). For a group of camera sensors, we divide their FoVs into several disjoint regions, where each region belongs to the FoVs of the same set of cameras. And then the entropy of each region can be estimated from the spatial correlation coefficients between cameras based on our spatial correlation model. The joint entropy is the sum of the entropies of all the regions. The EDM is verified through image coding experiments using commercial video coding standards.

With the joint entropy as a prediction of joint coding efficiency, we develop a

Distributed Multi-cluster Coding Protocol (DMCP) to partition the entire network into a set of coding clusters such that the global coding gain is maximized. The DMCP works in a fully distributed manner. It does not involve much communication cost, since each camera sensor just need to exchange its parameters such as sensing direction and location with its one-hop neighbors, from which the joint entropy of neighboring cameras can be estimated. Since the DMCP is a heuristic solution, we also analyze its performance and derive its approximation ratio compared to an optimal coding clustering problem.

1.2.3 Correlation-Based Scheduling

Network scheduling schemes have significant impact on the performance of sensor networks. To support QoS requirements (such as delay constraints) for WMSNs, contention-free scheduling is preferred in the MAC layer. We consider a clustered network architecture where camera sensors within a cluster follows contention-free TDMA scheduling. Under this architecture, we aim to utilize the correlation of visual information to boost the performance of scheduling in WMSNs.

We propose a differential coding based scheduling framework for WMSNs that consists of three components: MinMax Degree Hub Location (MDHL), Minimum Sum-entropy Camera Assignment (MSCA), and Maximum Lifetime Scheduling (MLS). The MDHL problem aims to find the optimal locations to place the multimedia processing hubs, which operate on different channels for concurrently collecting images from adjacent cameras, such that the number of channels required for frequency reuse is minimized. With the locations of the hubs determined by the MDHL problem, the objective of the MSCA problem is to assign each camera to a hub in such a way that the global compression gain is maximized by jointly encoding the correlated images gathered by each hub. At last, given a hub and its associated cameras, the MLS

problem targets at designing a schedule for the cameras such that the network lifetime is maximized by letting highly correlated cameras perform differential coding on the fly. We give performance bounds of these three problems, and then propose approximation and heuristic algorithms to solve them.

1.2.4 Correlation-Aware QoS Routing

Since camera sensors usually have large sensing radius, sensors that are out of the communication ranges of each other can still observe a common scene [58], (i.e., they are correlated with each other). This motivates us to utilize the correlation of camera sensors in the network layer. We can introduce correlation-aware operations in the routing process. Moreover, routing protocols for WMSNs should address the challenge of providing QoS support for various applications under the resource constraints of sensors. Considering these factors, we propose to jointly design the correlation-aware operations and QoS routing such that the energy consumption could be minimized while satisfying QoS requirements.

Based on the correlation of camera sensors, we first design a correlation-aware differential coding operation. In this operation, we evaluate the gains and costs for performing differential coding along routing paths, and let a sensor find a downstream node with the maximum gain to perform differential coding. For the correlated streams that cannot be further compressed, the presence of traffic congestion becomes evident in that video sensors with large overlapped field of views tend to report the same event and generate traffic concurrently. To solve this problem, we introduce a correlation-aware load balancing operation. The basic idea is to split the correlated flows that cannot be further compressed to different paths so that the probability of network congestion could be reduced. Finally, we integrate the correlation-aware operations in an optimization routing framework to minimize energy consumption while providing QoS support in the timeliness and reliability domains.

1.3 Organization of the Thesis

This thesis is organized as follows.

In Chapter 2, we present the spatial correlation model for visual information in WMSNs. We introduce the derivation of the spatial correlation coefficient first, followed by the entropy-based framework that estimates the amount of information from multiple correlated camera sensors. In the last part of this Chapter, we introduce the design of the correlation-based camera selection algorithm.

In Chapter 3, we present the collaborative image compression framework that maximizes the overall compression gain of visual information gathered in WMSNs. This framework consists of the Entropy-based Divergence Measure (EDM) that predicts the efficiency of joint coding among correlated camera sensors, and the Distributed Multi-cluster Coding Protocol (DMCP) that partitions a WMSN into a set of clusters to perform joint source coding.

In Chapter 4, we describe the correlation-based scheduling framework for WMSNs. We explain the three components of the scheduling framework in details: the MinMax Degree Hub Location (MDHL) problem that finds the optimal locations to place multimedia processing hubs, the Minimum Sum-entropy Camera Assisgnment (MSCA) problem that assigns camera sensors to processing hubs, and the Maximum Lifetime Scheduling (MLS) problem that finds the optimal schedule of camera sensors by enabling correlated camera sensors to perform differential coding in the scheduling process.

In Chapter 5, we present the correlation-aware QoS routing algorithm. We start by describing the correlation-aware operations: the correlation-aware inter-node differential coding scheme and the correlation-aware load balancing scheme. We then integrate the correlation-aware operations into a distributed QoS routing framework that minimizes energy consumption while providing probabilistic QoS guarantee for WMSN applications.

Finally, in Chapter 6, we draw the main conclusions and outline future research directions.

CHAPTER II

A SPATIAL CORRELATION MODEL FOR VISUAL INFORMATION IN WMSNS

2.1 Introduction

Since uncompressed raw video streams require excessive bandwidth that is impossible to be supported by wireless multihop networks, multimedia source coding must be employed to achieve high compression efficiency. Today's standardized video coding technologies, such as MPEG and H.26x [69], achieve high compression performance at the expense of extensive computation at the encoder. In contrast, distributed video coding [72] allows simple and low power encoder, while the decoder is high power and loaded with extensive computation burden. Current distributed video coding technologies rely on channel coding to exploit the correlation among adjacent frames [23, 51]. However, it is not easy to attain accurate estimations of the correlation structure among adjacent video frames, resulting in limited encoding efficiency of distributed video coding.

Except for multimedia source coding, collaborative multimedia in-network processing is suggested as an effective way to avoid the transmission of redundant information [5]. According to the requirements of specific applications, each sensor node can filter out uninteresting events locally, or coordinate with each other to aggregate correlated data. To design filtering and aggregation algorithms for WMSNs, the correlation characteristics of visual information from different sensors need to be studied.

In [61], a theoretical spatio-temporal correlation model is developed for scalar data in wireless sensor networks. However, as visual information is much more complex

than scalar data, the model in [61] cannot be directly applied to visual information.

In many recent research efforts for WMSNs, image processing techniques are utilized to design collaborative processing. In [62], images from correlated views are roughly registered using correspondence analysis. Each sensor transmits a low-resolution version of a common area, and the sink combines multiple low-resolution versions into a high-resolution image. In [71], spatial correlation is obtained by an image shape matching algorithm, while temporal correlation is calculated via background subtraction. Based on the spatial and temporal correlation, images from correlated sensors are transmitted collaboratively. However, the performance of image processing algorithms are application dependent: different types of images will require different processing schemes [25]. Also, image processing techniques are complicated and computation extensive, which will bring about extra computation costs for sensor nodes.

Cameras are directional sensors with limited field of views [42], and the image observed by a camera is directly related to its field of view. In [42], the correlation degree of two cameras is defined as the ratio of the overlapped sensing area to the entire area of one camera's field of view. A video processing scheme based on correlation is also proposed in [42]: two sensors cooperate with each other, and each sensor transmits a part of its observed image to the sink, and then the sink will combine the partial images together. But this scheme is only valid when the sensing directions of the two sensors do not differ very much. Besides, this processing method is limited between two sensors. How to deal with the cooperative processing of more than two sensors is a problem that has not been well investigated.

In this chapter, we study the correlation characteristics of visual information in WMSNs. Rather than using any specific image algorithms, we propose a general correlation model for visual information in WMSNs. Our main contributions are summarized as follows.

1. We design a novel *spatial correlation coefficient* to describe the correlation characteristics for the images observed by cameras with overlapped field of views. The spatial correlation coefficient will allow the estimation of correlation through low computation and communication costs in WMSNs.

2. We propose an *entropy-based analytical framework* to evaluate the joint effect of multiple correlated camera nodes, in which the joint entropy of multiple camera sensors is estimated as a function of the spatial correlation coefficients among the camera sensors.

3. Based on the entropy-based framework, we introduce a *correlation-based camera selection* algorithm that maximizes the gain of information for a certain number of sensors reporting to the sink.

2.2 Problem Statement

In a multimedia sensor network, multiple camera sensors are deployed to provide multiple views, multiple resolutions, and enhanced observations of the environment [14]. Fig. 1 gives an example of a WMSN deployed with cameras. A typical scenario of WMSN is: the application specifies which area it is interested in, and the cameras that can observe this area will report their observations to the sink. For a certain area of interest, suppose there are N camera sensors that can observe it. We denote them as a group $S = \{S_1, S_2, ..., S_N\}$, and their observed images as $\{X_1, X_2, ..., X_N\}$. There exists correlation among the observations of this group of cameras, which can be exploited to design multimedia in-network processing schemes.

2.2.1 Spatial Correlation

First, we study the correlation characteristics of the images observed by different cameras. For Camera i and Camera j in the group $S = \{S_1, S_2, ..., S_N\}$, we will derive a correlation coefficient ρ_{ij} to describe the degree of correlation between image

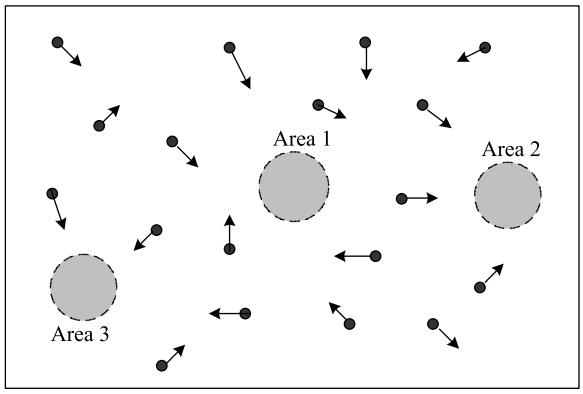

Figure 1: WMSN with camera sensors and areas of interest.

X_i and image X_j. For the group of camera sensors, the correlation among the images observed by these cameras ($\{X_1, X_2, ..., X_N\}$) will be represented as a correlation matrix C, denoted as $C = (\rho_{ij})_{N*N}$, where ρ_{ij} is the correlation coefficient of image X_i and image X_j.

2.2.2 Joint Effect of Multiple Correlated Cameras

After we obtain the spatial correlation coefficient, we study the joint effect of multiple correlated cameras in WMSNs. In particular, we study how to measure the amount of visual information from multiple cameras in a WMSN. Intuitively, the visual information provided by multiple cameras should be related to the correlation characteristics of the observed images. If the images observed by these cameras are less correlated, they will provide more information to the sink. We develop an entropy-based framework to estimate the amount of information from multiple correlated cameras.

2.2.3 Correlation-based Camera Selection

Since the delivery of visual information needs very high bandwidth, which may reduce the lifetime of the network, the communication load in WMSNs should be reduced as much as possible. Suppose a total number of N cameras can observe the area of interest, if network resources permit, we can let all these N cameras transmit their observed images to the sink, so that the users at the sink can obtain comprehensive

information about the area. However, if the sink/application allows a certain level of distortion of the observations, it may not be necessary for all the cameras to report their observed information to the sink.

Consequently, we define a camera selection problem as follows: if only M cameras ($M \leq N$) are allowed to transmit their observed images to the sink, how to select M cameras out of the N cameras so that the sink can gain the maximum amount of information. Based on our study on the joint effect of multiple cameras, we design a correlation-based algorithm to select cameras under distortion constraints.

2.3 Spatial Correlation Model for Visual Information

2.3.1 Sensing Model

Different from scalar data sensors, the sensing of a camera is characterized by directional sensing and 3-D to 2-D projection. In computer vision, this sensing process is usually described by the pinhole camera model [21]. Fig. 2 illustrates an example of a pinhole camera. The camera's center of projection is at the origin of a Euclidean coordinate system, and its sensing direction is along the x axis. The focal length of the camera is f, so the image plane is the plane $x = f$. A scene point P with coordinates $(x, y, z)^T$ is mapped to $P'(u, v)^T$ on the image plane, where u and v are given by

$$\begin{cases} u = fy/x \\ v = fz/x \end{cases} \tag{1}$$

A camera also has limited sensing range. It can only observe the objects within its field of view (FoV). A simplified 2-D FoV model is proposed in [42]: as shown in Fig. 3, a camera's field of view is determined by four parameters $(P, R, \vec{V}, \alpha)$, where P is the location of the camera, R is the sensing radius, $\vec{V}$ is the sensing direction (the center line of sight of the camera's field of view), and α is the offset angle.

A camera's focal length can be estimated by various calibration methods [21].

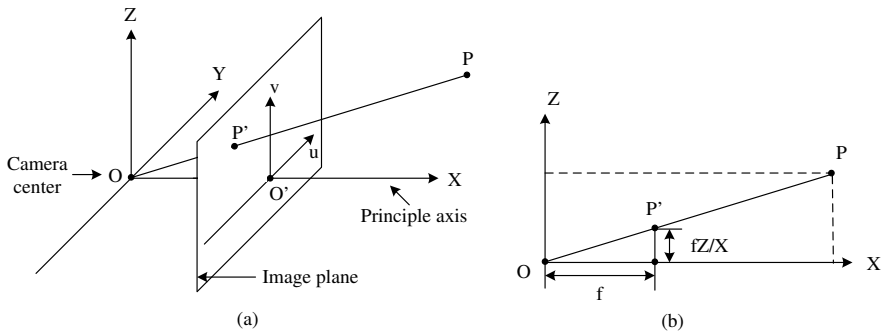

Figure 2: Camera projection model.

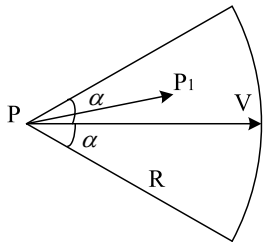

Figure 3: Field of view.

More recently, several methods have been proposed for the calibration and localization of cameras in sensor networks [9, 18]. Each camera's focal length (f), location (P), and sensing direction ($\vec{V}$) can be estimated as shown in [18]. In the following analysis, we will derive a spatial correlation function based on these parameters.

2.3.2 System Model

We set up a world coordinate system $(W) = (O_W, i_W, j_W, k_W)$ for the area of interest as shown in Fig. 4(a), in which the origin is the center of the area of interest, and the XOY plane is the ground plane. Seven reference points, which can also be regarded as *feature points* or key points in a scene, are chosen as: $O(0,0,0)^T$, $A(1,0,0)^T$, $B(-1,0,0)^T$, $C(0,1,0)^T$, $D(0,-1,0)^T$, $E(0,0,1)^T$, $F(0,0,-1)^T$. These reference points form six unit reference vectors along the orthogonal directions in the 3-D world: $\vec{OA}$, $\vec{OB}$, $\vec{OC}$, $\vec{OD}$, $\vec{OE}$, $\vec{OF}$.

We consider the case when all the camera sensors are placed on the ground plane (XOY) and their sensing directions are also within the ground plane. For a camera

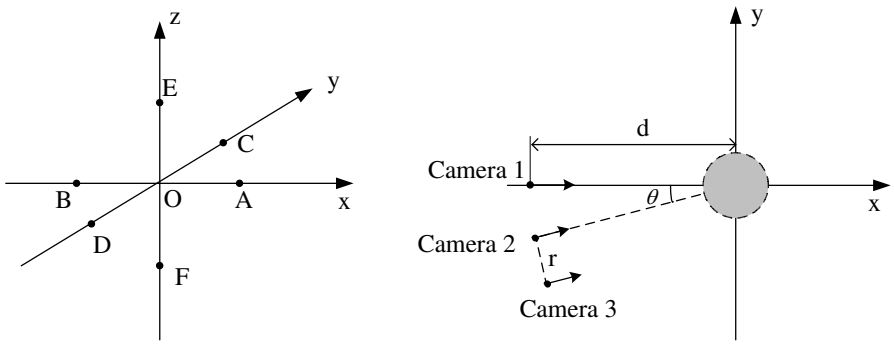

Figure 4: Reference points in the area of interest and deployment of cameras.

sensor S_i, the coordinates of its optical center can be denoted as $(x_i, y_i)^T$. The sensing direction of S_i can be described by a unit vector $\phi(\phi_x, \phi_y)$, where $\phi_x = cos\theta$, $\phi_y = sin\theta$, and θ is the angle between the sensing direction and the x axis.

The projections of the reference points on a camera will change as the camera's location and sensing direction change. By comparing the projections of the same reference points at different cameras, we can understand the correlation characteristics among different cameras.

2.3.3 Projection Geometry

Fig. 4(b) shows the deployment of three cameras in the world coordinate system (W), where the origin is the center of the area of interest, and the XOY plane denotes the ground plane. Camera 1 is located at $(-d, 0)^T$ with its sensing direction along the x axis. Camera 2 is located at $(-dcos\theta, -dsin\theta)^T$, and its sensing direction rotates an angle of θ about the x axis. For both Camera 1 and Camera 2, their principle axes pass the center of the area of interest (the origin). Camera 3 has the same sensing direction as Camera 2, but its principle axis does not pass the origin. The distance from the center to its principle axis is r, as shown in Fig. 4(b). The optical center of Camera 3 is $(-dcos\theta + rsin\theta, -dsin\theta - rcos\theta)^T$. Although the locations and sensing directions of these three cameras are different, the depths for the center of the area

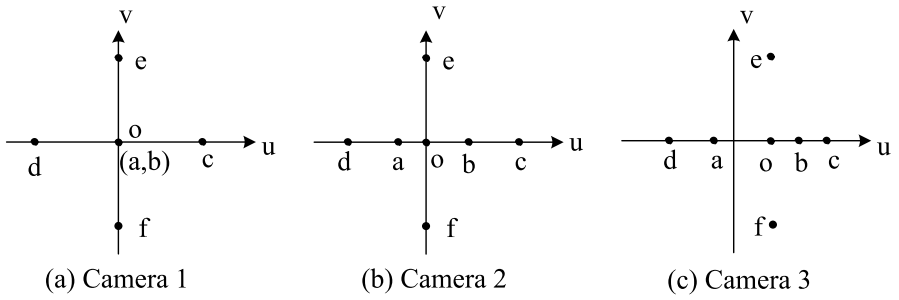

Figure 5: Projections of reference points and vectors.

of interest in all the three cameras have the same value d. In addition, we assume that all these cameras have the same focal length f.

To calculate the projections of the reference points in a camera, a *coordinate transform* is first needed to obtain the coordinates of the points in the camera's coordinate system. For example, the coordinate system of Camera 1 (Fig. 2) and the world coordinate system (W) are separated by a pure translation. For an arbitrary point P in the space, we have

$$^1P = ^W P + ^1 O_W, \tag{2}$$

where ^1P is the coordinate vector of point P in the coordinate system of Camera 1, while ^WP is the coordinate vector in the world coordinate system (W). These two vectors are related by ^1O_W, the coordinate vector of the origin in (W) seen in the coordinate system of Camera 1. Here $^1O_W = (d, 0, 0)^T$.

Therefore, the coordinates of the reference points in the coordinate system of Camera 1 are as follows: $O_1(d,0,0)^T$, $A_1(d+1,0,0)^T$, $B_1(d-1,0,0)^T$, $C_1(d,1,0)^T$, $D_1(d,-1,0)^T$, $E_1(d,0,1)^T$, $F_1(d,0,-1)^T$.

Based on the projection model in equation (1), we can find the projections of these reference points in Camera 1: $o_1(0,0)^T$, $a_1(0,0)^T$, $b_1(0,0)^T$, $c_1(\frac{f}{d},0)^T$, $d_1(-\frac{f}{d},0)^T$, $e_1(0,\frac{f}{d})^T$, $f_1(0,-\frac{f}{d})^T$. The projections of reference points on Camera 1 are plotted in Fig. 5(a).

As for Camera 2, its coordinate system can be derived from the world coordinate

system (W) as follows: rotate the world coordinate system counterclockwise for an angle of θ, and then translate the rotated system along the negative direction of x axis for a length of d, given as

$$^2P = \ ^2_W R^W P + \ ^2O_W, \tag{3}$$

where ^2O_W is the translation offset vector, $^2O_W = (d, 0, 0)^T$, and $^2_W R$ is the rotation matrix,

$$^2_W R = \begin{pmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Similarly, the relationship between the coordinate system of Camera 3 and the world coordinate system is related as

$$^3P = \ ^3_W R^W P + \ ^3O_W, \tag{4}$$

where the rotation matrix is the same as that of Camera 2, $^3_W R = \ ^2_W R$, while the translation offset vector satisfies $^3O_W = (d, r, 0)^T$.

As the case of Camera 1, the projections of reference points on Camera 2 and Camera 3 can be calculated in the same way. Table 1 lists the projections of the seven reference points on the three cameras, and Fig. 5 illustrates the positions of reference points on the three cameras. Based on the coordinates of the reference points, the values of the corresponding unit vectors are also calculated, which are listed in Table 2.

2.3.4 Spatial Correlation Coefficient

2.3.4.1 Scaling Effect

Comparing the projections of the reference vectors in Table 2, we find that the lengths of OA, OB, OC, and OD are different in the three cameras, but the length of OE/OF remains to be a constant value. The reason is that the points O, E, and F have the

Table 1: Projections of reference points

Points	Projections		
	Camera 1	Camera 2	Camera 3
O	$(0,0)^T$	$(0,0)^T$	$(\frac{r}{d}f,0)^T$
A	$(0,0)^T$	$(\frac{-sin\theta}{d+cos\theta}f,0)^T$	$(\frac{r-sin\theta}{d+cos\theta}f,0)^T$
B	$(0,0)^T$	$(\frac{sin\theta}{d-cos\theta}f,0)^T$	$(\frac{r+sin\theta}{d-cos\theta}f,0)^T$
C	$(\frac{f}{d},0)^T$	$(\frac{cos\theta}{d+sin\theta}f,0)^T$	$(\frac{r+cos\theta}{d+sin\theta}f,0)^T$
D	$(-\frac{f}{d},0)^T$	$(\frac{-cos\theta}{d-sin\theta}f,0)^T$	$(\frac{r-cos\theta}{d-sin\theta}f,0)^T$
E	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$	$(\frac{r}{d}f,\frac{f}{d})^T$
F	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$	$(\frac{r}{d}f,-\frac{f}{d})^T$

Table 2: Projections of reference vectors

Vectors	Projections		
	Camera 1	Camera 2	Camera 3
$\vec{OA}$	$(0,0)^T$	$(\frac{-sin\theta}{d+cos\theta}f,0)^T$	$(\frac{r-sin\theta}{d+cos\theta}f-r\frac{f}{d},0)^T$
$\vec{OB}$	$(0,0)^T$	$(\frac{sin\theta}{d-cos\theta}f,0)^T$	$(\frac{r+sin\theta}{d-cos\theta}f-r\frac{f}{d},0)^T$
$\vec{OC}$	$(\frac{f}{d},0)^T$	$(\frac{cos\theta}{d+sin\theta}f,0)^T$	$(\frac{r+cos\theta}{d+sin\theta}f-r\frac{f}{d},0)^T$
$\vec{OD}$	$(-\frac{f}{d},0)^T$	$(\frac{-cos\theta}{d-sin\theta}f,0)^T$	$(\frac{r-cos\theta}{d-sin\theta}f-r\frac{f}{d},0)^T$
$\vec{OE}$	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$	$(0,\frac{f}{d})^T$
$\vec{OF}$	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$	$(0,-\frac{f}{d})^T$

same depth (d) in all the three cameras, and that the cameras also have the same focal length (f). Both the depth and the focal length can influence the size of a projection. Thus, we define a scaling factor, s, as the lengths of the projections of OE and OF, given by

$$s = \frac{f}{d}. \tag{5}$$

2.3.4.2 Translation Effect

As can be seen in Fig. 5, the projections on Camera 1 and Camera 2 are both in the center of the image planes, but the projections on Camera 3 have an offset from the center of the image plane. The deviation from the center of the area of interest to the camera's principle axis has caused the translation of the projections. Based on the projections of reference points on Camera 3, we define a translation factor as

$$t = r\frac{f}{d} = rs. \tag{6}$$

2.3.4.3 Correlation Coefficient

As shown in Table 2, the lengths of vectors OA, OB, OC, and OD will change as the camera's location and sensing direction change. Based on this observation, we design a *disparity function* to reveal the disparity between the projections of reference vectors on different cameras. Suppose that Camera i and Camera j are two arbitrary cameras on the ground plane that can observe the area of interest, the disparity function is derived as below:

1. Determine the positions and sensing directions of Camera i and Camera j;
2. Based on the projection model in (1), compute the projections of reference vectors in each camera;
3. Divide the projections of reference vectors by the scaling factor $s = \frac{f}{d}$ (5), so that we can get a set of *normalized projection vectors* for each camera;

4. Compute the distance for each pair of normalized vectors OA, OB, OC, and OD. For example, if the projection of OA is $o_i a_i = (u_i, v_i)^T$ on Camera i, and $o_j a_j = (u_j, v_j)^T$ on Camera j , the distance is calculated as

$$d_{OA} = \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2}; \tag{7}$$

5. The disparity between the images at Camera i and Camera j, denoted by δ, is defined as the average distance of the four vectors:

$$\delta = \frac{1}{4}(d_{OA} + d_{OB} + d_{OC} + d_{OD}). \tag{8}$$

For Camera 1 and Camera 2 in Fig. 4(b), according to the results in Table 2, the disparity between the images at Camera 1 and Camera 2 is calculated as

$$\delta = \frac{1}{4}(|\frac{d\sin\theta}{d+\cos\theta}| + |\frac{d\sin\theta}{d-\cos\theta}| + |\frac{d\cos\theta}{d+\sin\theta} - 1| + |\frac{-d\cos\theta}{d-\sin\theta} + 1|). \tag{9}$$

Generally, for Camera i and Camera j with position parameters (d_i, r_i, θ_i) and (d_j, r_j, θ_j) (Fig. 4(b)), the disparity between the images at the two cameras is given by

$$\begin{aligned}\delta_{ij} = \frac{1}{4}(&|\frac{-d_i\sin\theta_i - r_i\cos\theta_i}{d_i+\cos\theta_i} - \frac{-d_j\sin\theta_j - r_j\cos\theta_j}{d_j+\cos\theta_j}| + \\ &|\frac{d_i\sin\theta_i + r_i\cos\theta_i}{d_i-\cos\theta_i} - \frac{d_j\sin\theta_j + r_j\cos\theta_j}{d_j-\cos\theta_j}| + \\ &|\frac{d_i\cos\theta_i - r_i\sin\theta_i}{d_i+\sin\theta_i} - \frac{d_j\cos\theta_j - r_j\sin\theta_j}{d_j+\sin\theta_j}| + \\ &|\frac{-d_i\cos\theta_i + r_i\sin\theta_i}{d_i-\sin\theta_i} - \frac{-d_j\cos\theta_j + r_j\sin\theta_j}{d_j-\sin\theta_j}|).\end{aligned} \tag{10}$$

We present a simulation to show how the disparity value δ varies as a function of the deployments of cameras. Refer to Fig. 4(b), we let Camera 1 stay fixed, and let the sensing direction of Camera 2 (θ) change from $-90°$ to $90°$. The *sensing direction difference* between Camera 2 and Camera 1 is also θ. Set the depth $d = 2.5(meters)$ for Camera 1 and Camera 2. The disparity between Camera 1 and Camera 2 in (9) is illustrated as a function of θ (in degrees) in Fig. 6.

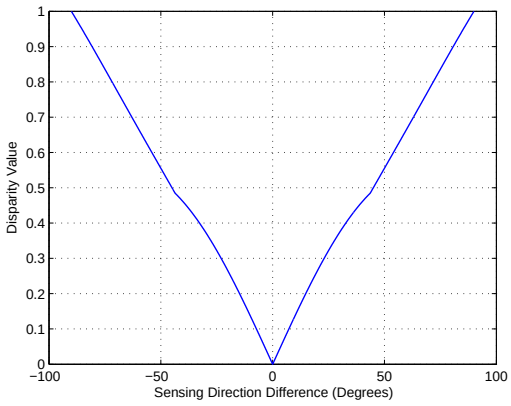

Figure 6: Illustration of the disparity function.

The disparity value increases as the sensing direction difference increases. The larger the disparity value, the more differences exist between the two images, i.e. the images are less correlated. In the above scenario, the largest disparity value goes to 1 when the sensing directions of the two cameras are perpendicular, for which we can say that the two cameras are weakly correlated. For the convenience of further analysis, we bound the disparity value from 0 to 1 as follows:

$$\delta = min(\delta, 1). \tag{11}$$

Consequently, we can define a *correlation coefficient* that is complementary to the disparity function:

$$\rho = 1 - \delta. \tag{12}$$

When the correlation coefficient is 0, it means that the two images are independent of each other. If it equals to 1, the two images are highly correlated. The larger the correlation coefficient, the more correlated are the two images.

2.3.5 Discussion

In WMSN applications, as long as the area of interest is specified, and the locations and sensing directions of cameras are estimated, the correlation characteristics of

cameras with overlapped field of views can be obtained as introduced above. The proposed correlation model can help to design the differential source coding between cameras, as well as the aggregation of visual information in the network.

The proposed model depends on the selection of reference points/vectors in the area of interest. Six unit vectors along three orthogonal directions in the 3-D world are chosen in the above analysis. For a WMSN application, the reference points should be chosen properly based on specific application requirements. In addition, a camera's field of view will be reduced when it is blocked by some obstacles. To guarantee that our model works well, a camera's practical field of view needs to be estimated.

2.4 Joint Effect of Multiple Correlated Cameras

In this section, we study the joint effect of multiple correlated cameras. We investigate how to measure the amount of visual information from multiple correlated cameras and then propose a correlation-based camera selection algorithm.

2.4.1 Entropy-based Approach

In information theory [12], the concept of entropy is used to measure the amount of information of a random source. If an image is interpreted as a sample of a "gray-level source", the source's symbol probabilities can be modeled by the gray-level histogram of the observed image. An estimate of the source's entropy can be generated as [25]

$$\widetilde{H} = -\sum_{k=1}^{L} p(r_k) log p(r_k), \tag{13}$$

where L is the number of all possible gray-levels, and $p(r_k)$ is the probability of the kth gray-level. It denotes the average amount of information per pixel in the image.

If a camera S_i transmits its observed image X_i to the sink, the amount of information gained at the sink is $H(X_i)$. (We do not consider the information loss caused by

lossy compression or packet loss during transmission.) If the group of camera sensors, $S = \{S_1, S_2, ..., S_N\}$, transmit their observed images $\{X_1, X_2, ..., X_N\}$ to the sink, the amount of information gained at the sink will be the joint entropy $H(X_1, X_2, ..., X_N)$. Our objective is to estimate the joint entropy of multiple cameras.

2.4.2 Joint Entropy of Two Cameras

We consider two cameras that can observe the area of interest. Suppose each camera has captured one image about the area of interest, denoted as image A and image B. The joint entropy of A and B is

$$H(A, B) = H(A) + H(B) - I(A; B), \tag{14}$$

where $I(A; B)$ is the mutual information of the two sources. $I(A; B)$ can be interpreted as the reduction in the uncertainty of one source due to the knowledge of the other source:

$$I(A; B) = H(A) - H(A|B) = H(B) - H(B|A). \tag{15}$$

The definition of $I(A; B)$ in probability form is given as

$$I(A; B) = \sum_a \sum_b p(a, b) log \frac{p(a, b)}{p(a)p(b)}. \tag{16}$$

where $p(a)$ and $p(b)$ are the probability distributions of the pixels in image A and image B, and $p(a, b)$ is the joint probability distribution of the two sources.

Mutual information is a measure of dependence between two sources: the more A and B are correlated, the larger the mutual information $I(A; B)$.

In [50], a normalized form of mutual information, *entropy correlation coefficient (ECC)*, is defined as

$$ECC = \frac{2I(A; B)}{H(A) + H(B)}. \tag{17}$$

The entropy correlation coefficient (ECC) ranges from zero to one, where zero indicates that source A and B are independent, while one indicates that source A

equals to source B. The larger the ECC value, the more these two sources are correlated.

Based on (14) and (17), the joint entropy of A and B can be expressed as a function of $H(A)$, $H(B)$ and ECC:

$$H(A,B) = (1 - \frac{1}{2}ECC)(H(A) + H(B)). \tag{18}$$

Since $H(A)$ and $H(B)$ can be calculated at each camera using (13), if ECC can be estimated, the joint entropy $H(A,B)$ will be obtained. However, to calculate $I(A;B)$ and ECC, a joint probability distribution of the two sources needs to be estimated (16). Due to the complexity of image contents and the difficulty in image modeling, it is difficult to get an accurate estimation of the joint probability distribution [50]. Besides, estimating the joint probability also requires large bulk of computation [50]. If joint probability distribution is to be estimated in a sensor network, cameras at different locations must exchange their observed images, which will introduce a lot of communication burden in the network.

It can be seen that the proposed correlation coefficient in (12) has the same intrinsic meaning as ECC: both ranging from 0 to 1 and denoting the degree of correlation between two sources. However, if cameras' parameters and deployment information are given, it is much easier to obtain the proposed correlation coefficient. Considering the limited processing capability of sensors, we propose to estimate ECC by the proposed correlation coefficient. If we replace ECC in (18) by the proposed correlation coefficient ρ, we can obtain an estimation of the joint entropy of A and B as

$$H(A,B) \approx (1 - \frac{1}{2}\rho)(H(A) + H(B)). \tag{19}$$

Therefore, the amount of information that can be gained from image A and image B together depends on the correlation degree between A and B. The more A and B are correlated, the less joint entropy can be gained from A and B together. That

is to say, if two camera sensors transmit their images to the sink, the amount of information gained at the sink will be larger if the two sensors are less correlated.

2.4.3 Joint Entropy of Multiple Cameras

In this section, we extend our study of joint entropy to the case of more than two cameras. Suppose there is a group of camera sensors $S = \{S_1, S_2, ..., S_N\}$ with their observed images $\{X_1, X_2, ..., X_N\}$. We are interested in estimating the joint entropy $H(X_1, X_2, ..., X_N)$ for this group of sensors. If $H(X_1, X_2, ..., X_N)$ is to be computed by its definition in probability, the joint probability distribution of these N images needs to be estimated. However, it is difficult to estimate the joint probability distribution of multiple sources, especially when N is large.

A feasible approach is to make use of the joint entropy of two cameras in the last section. As there are N individual elements in the group $\{X_1, X_2, ..., X_N\}$, we can merge two of them together, so that the joint entropy of these two elements can be calculated by (19). We treat these two elements as a whole element, then the number of elements in the group reduces to $N - 1$. If we repeat this process, the N individual sensors will be combined into a single element in the end. As the joint entropy of merged sensors are calculated along the merging process, the joint entropy $H(X_1, X_2, ..., X_N)$ can be obtained when the merging process is completed.

We design an algorithm to estimate the joint entropy of multiple cameras based on the idea of hierarchical clustering [33]. As long as the entropy of each single image $(H(X_i), i = 1, 2, ..., N)$ and the correlation matrix $(C = (\rho_{ij})_{N*N})$ are given, the joint entropy $H(X_1, X_2, ..., X_N)$ can be estimated through the hierarchical clustering process. The details of the estimation algorithm are presented in *Algorithm 1*, where χ denotes the set of clusters, and $\rho(\{X_i\}, \{X_j\})$ is the correlation coefficient between cluster $\{X_i\}$ and cluster $\{X_j\}$.

Algorithm 1 Estimate the Joint Entropy of Multiple Cameras

$H(X_1, X_2, ..., X_N) = JointEntropy(H(X_i), (\rho_{ij})_{N*N})$
begin
$\chi = \{\{X_1\}, \{X_2\}, ..., \{X_N\}\}$, $\rho(\{X_i\}, \{X_j\}) = \rho_{ij}$.
for $k = 1$ to $N - 1$ **do**
  Find $(\{X_i\}, \{X_j\}) = arg \max_{\{X_i\}, \{X_j\} \in \chi} \{\rho(\{X_i\}, \{X_j\})\}$ {Find the most correlated pair of clusters in χ.}
  Merge $\{X_i\}$ and $\{X_j\}$ into a new cluster $\{X_{N+k}\}$.
  $H(X_{N+k}) = H(X_i, X_j)$ (19).
  for $X_l \in \chi, l \neq i, l \neq j$ **do**
    Compute $\rho(\{X_{N+k}\}, \{X_l\})$. $(*)$
  end for
  Remove $\{X_i\}$ and $\{X_j\}$ from χ; Add the new cluster $\{X_{N+k}\}$ into χ.
end for
$H(X_1, X_2, ..., X_N) = H(X_{2N-1})$
return $H(X_1, X_2, ..., X_N)$
end

In step $(*)$ of *Algorithm 1*, the correlation coefficient between one cluster and another cluster can be obtained by the *greatest/shortest/average* correlation coefficient from any member of one cluster to any member of the other cluster [33], which are referred to as *single-linkage/complete-linkage/average-linkage* clustering.

The following is an example of the estimation of joint entropy. Suppose there is a group of five camera sensors. Without loss of generality, we assume that the entropy of a single image is a constant value, denoted as $H(X_i) = H(\cdot)(i = 1, ..., 5)$. A correlation matrix for these five sensors is given by

$$(\rho_{ij})_{5*5} = \begin{pmatrix} 1 & 0 & 0.2942 & 0.9443 & 0 \\ & 1 & 0 & 0 & 0.7359 \\ & & 1 & 0.3416 & 0 \\ & & & 1 & 0 \\ & & & & 1 \end{pmatrix}. \tag{20}$$

Apply *Algorithm 1* to this group of sensors, and use the *average-linkage clustering* [33] metric in step $(*)$. The clustering process is illustrated in Fig. 7, and the results in each step of clustering are shown in Table 3. Comparing the values in the correlation

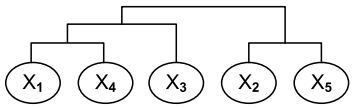

Figure 7: An example of hierarchical clustering.

Table 3: Hierarchical clustering steps

Steps	Nodes for Clustering	Estimation of joint entropy (Relative value to $H(\cdot)$)
1	$\{X_1\}$,$\{X_4\}$	$H(X_1, X_4) = 1.0557$
2	$\{X_2\}$,$\{X_5\}$	$H(X_2, X_5) = 1.2641$
3	$\{X_1X_4\}$,$\{X_3\}$	$H(X_1, X_3, X_4) = 1.7290$
4	$\{X_1X_3X_4\}$,$\{X_2X_5\}$	$H(X_1, X_2, X_3, X_4, X_5) = 2.9931$

matrix (20) and the clustering steps in Fig. 7, one can find that in every clustering step, nodes that contain the most correlated images are merged into one cluster. As can be seen from (19), the value of the joint entropy decreases as the correlation degree of the two images increases. Therefore, the joint entropies obtained from the clustering process are always relatively small. The final result of the estimation algorithm is a conservative estimation of joint entropy.

2.4.4 Correlation-based Camera Selection

Suppose for an area of interest in a WMSN, a total number of N cameras can observe the area of interest. If network resources permit, all these cameras can transmit their observed images to the sink, so that the applications at the sink can gain comprehensive information about the area. However, as the processing capabilities of sensors are limited, and the communication among sensors causes huge energy consumption, sometimes the network cannot support all these cameras to report their observations to the sink. Consequently, we define a camera selection problem: if only M cameras ($M \leq N$) are allowed to transmit their observed images to the sink, how to select M cameras out of the N cameras so that the sink can gain the maximum amount of information.

As in the last section, we also assume that the entropy of a single image is a constant value here. The estimation of joint entropy in (19) indicates that the less correlated are the two sensors, the more information can be provided by the two sensors together. Thus, to maximize the joint entropy of M cameras, we should try to minimize the correlation among the cameras to be selected. We propose a *correlation-based* algorithm to maximize the joint entropy of M cameras. At each step of the algorithm, we select one camera that is least correlated with the cameras that have already been selected. The details are presented in *Algorithm 2*, where $\chi = \{X_1, X_2, ..., X_N\}$ is the set of images observed by these N cameras, and S denotes the set of cameras that are already selected.

Algorithm 2 Correlation-based Camera Selection

$S = CorrSelection(\{X_1, X_2, ..., X_N\}, (\rho_{ij})_{N*N}, M)$
begin
$S = \emptyset$, $\chi = \{X_1, X_2, ..., X_N\}$, $\rho(X_i, X_j) = \rho_{ij}$.
Find $(X_i, X_j) = arg \min_{X_i, X_j \in \chi} \{\rho(X_i, X_j)\}$ {Find the least correlated pair of cameras.}
Add the corresponding X_i and X_j into S. $\{M = 2\}$
if $M > 2$ **then**
  for $k = 1$ to $M - 2$ **do**
    for $X_l \in \chi, X_l \notin S$ **do**
      $\rho(X_l, S) = \max_{X_j \in S} \{\rho(X_l, X_j)\}$.
    end for
    $X_m = arg \min_{X_m \in \chi, X_m \notin S} \{\rho(X_m, S)\}$; Add X_m into S.
  end for
end if
return $S = \{X_{i1}, X_{i2}, ..., X_{iM}\}$
end

2.4.5 A Distortion Function

For an area of interest in WMSN, we suppose a total number of N cameras can observe it, and denote their observed images as $\{X_1, X_2, ..., X_N\}$. The joint entropy of all these N sensors, $H(X_1, X_2, ..., X_N)$, is the maximum amount of information that can be gained for the area of interest. If a subset of these sensors, denoted

Figure 8: Images. (a) $\theta = 0°$, (b) $\theta = 15°$, (c) $\theta = 30°$, (d) $\theta = 45°$, (e) $\theta = 60°$.

as $\{X_{i1}, X_{i2}, ..., X_{iM}\}$, are selected to report their observed images to the sink, the information gained at the sink is $H(X_{i1}, X_{i2}, ..., X_{iM})$.

We define a distortion function as the ratio of the decrease in the amount of information to the maximum amount of information, given by

$$D = \frac{H(X_1, X_2, ..., X_N) - H(X_{i1}, X_{i2}, ..., X_{iM})}{H(X_1, X_2, ..., X_N)}. \tag{21}$$

The value of D satisfies $0 \leq D \leq 1$. It can be interpreted as the percentage of information loss due to network resource constraints. Applications of WMSNs can use this distortion function as a metric to describe their requirements. For example, an application may ask the network to transmit information within 10% or 20 % of information loss.

It should be emphasized that the proposed distortion function is different from existing image/video quality metrics. Commonly used image quality metrics, such as PSNR (peak signal-to noise ratio) and the recently developed SSIM (structural similarity) [68], are designed to evaluate the degradation of a distorted image compared to an original image, where distortion is caused by lossy compression or loss during transmission.

However, our proposed distortion function is designed to evaluate the joint effect of multiple images. Distortion is the percentage of information loss caused by reporting a subset of images to the sink. According to the derivation of joint entropy in *Algorithm 1*, we can find that the value of distortion is related to the number of selected cameras as well as the correlation among the selected cameras.

2.5 Performance Evaluation

2.5.1 Spatial Correlation Coefficient

In this section, we present a set of experiments to evaluate the performance of our spatial correlation model.

2.5.1.1 Validity of the Proposed Spatial Correlation Coefficient

We set up a scene as shown in Fig. 4(b): Camera 1 and Camera 2 are placed to take pictures of an area of interest. Camera 1 is placed along the x axis, and Camera 2 rotates an angle of θ, so the *sensing direction difference* between Camera 2 and Camera 1 is θ. Set $d = 2.5(meters)$. A reference image is obtained at Camera 1, and then a group of 10 images are taken for Camera 2 with the following θ values: $\{-75°,-60°,-45°,-30°,-15°,15°,30°,45°,60°,75°\}$. Fig. 8 presents some of the images.

In Section 2.3, we showed that the degree of correlation is relevant to cameras' sensing directions and their relative positions. Since the sensing directions and positions are already known, the disparity between the test images on Camera 2 and Camera 1 can be easily calculated by function (10). The results of the disparity values are presented in Fig. 9(a) as a function of the *sensing direction difference* θ. The disparity increases as the sensing direction difference increases.

The correlation between images can also be obtained by applying image processing algorithms. Here, we refer to a commonly used feature extraction algorithm in [43]. In this algorithm, texture features are extracted from the images by Gabor wavelet transform, and based on the wavelet coefficients, feature vectors in multiple scales and multiple directions are constructed. Finally, an average distance is calculated by averaging all the feature distances in each feature space [43]. The average feature distances between the test images on Camera 2 and Camera 1 are calculated accordingly, and the results are presented in Fig. 9(b).

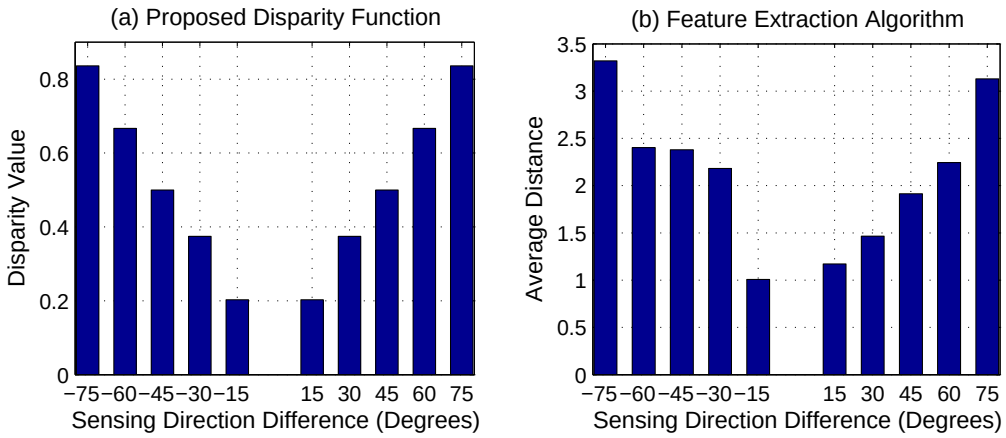

Figure 9: Proposed disparity function vs. feature extraction algorithm.

Comparing the results of the proposed model (Fig. 9(a)) and the results of the feature extraction algorithm in [43] (Fig. 9(b)), we find that in both cases the disparity/distance value increases as the sensing direction difference increases. This is also in accordance with our common sense: if we just observe the test images in Fig. 8 with our eyes, we can also find that the two images from Camera 2 and Camera 1 look more different when their sensing direction difference (θ) is larger. Therefore, the proposed spatial correlation coefficient is effective as it can reveal the correlation characteristics between images.

The slight differences between the results in Fig. 9(a) and Fig. 9(b) may be explained by the intrinsic differences of the two schemes. The proposed model is derived by studying cameras' sensing model and deployments, thus, the results are just dependent on a few parameters. In contrast, the feature extraction algorithm goes into details in an image. It is sensitive to the noise in the images, and even a little change of the light condition might influence the final results.

2.5.1.2 Costs for Exploiting Correlation

In this section, we discuss about the costs for exploiting correlation in WMSNs. As shown in the example of WMSN in Fig. 1, a typical scenario of WMSN application

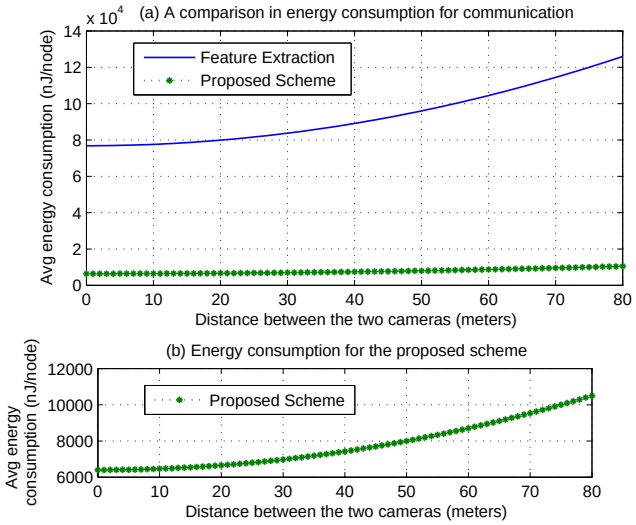

Figure 10: Average energy consumption for communication per node.

is: the application specifies which area it is interested in, and the cameras that can observe this area will work together to provide enhanced observations for the application. Given an area of interest, suppose a group of N cameras can observe it. If the cameras in this group want to know their correlation characteristics with each other, communication and computation operations are needed for these camera sensors. Note that it is a repetitive process to exploit correlation in the network: the correlation characteristics are obtained with respect to a certain area of interest, therefore, once the application specifies for a different area of interest, the correlation characteristics need to be investigated again.

We study a single hop case between two cameras, so that the results will be independent of specific communication protocols and network topologies. Assume two arbitrary cameras in the group, Camera i and Camera j, are within the transmission range of each other, and they will cooperate with each other to obtain the correlation of their observed images.

The proposed correlation model is derived based on the sensing model and deployment information of camera sensors. In most sensor networks, localization algorithms are already implemented, so that each camera knows its position in the network. The

focal length and sensing direction for each camera can be estimated [18] and recorded in the deployment stage of the network. Thus, when the application specifies a certain area of interest, each camera can easily figure out its position with respect to the area of interest (d, r, and θ as shown in Fig. 4(b)).

To calculate the correlation between Camera i and Camera j, Camera i just needs to transmit its four parameters to Camera j: d, r, θ as shown in Fig. 4(b), and its focal length f. Once Camera j receives the four parameters, it can calculate the correlation coefficient based on (12). The total energy consumption will be composed of the energy consumption for transmitting and receiving the four parameters and the energy consumption to calculate the correlation coefficient. It can be seen that the energy consumption for the proposed model is independent of image sizes.

We take the commonly used feature extraction scheme in [43] as a representative of the various image processing schemes. As we have introduced above, the feature extraction scheme [43] implements Gabor Wavelets to extract features vectors from multiple scales and multiple resolutions. If this scheme is implemented in sensor networks, Camera i will need to exchange its extracted features with Camera j to obtain the correlation degree between Camera i and Camera j. A typical process is as follows: Camera i extracts the features of its observed image using Gabor Wavelet, and transmits its feature vectors; Camera j receives the feature vectors from Camera i, and also implement the Gabor wavelet to extract the features of its own image. Finally, the correlation of images at Camera i and Camera j can be calculated by comparing their feature vectors.

The proposed scheme needs to transmit four parameters. As each parameter needs 32 bits to present, the total bits for transmission is 4*32 bits. In the feature extraction scheme [43], features are extracted from 4 resolutions and 6 orientations, and each feature space contains 2 elements. We also assume that each element in the feature space needs 32 bits to present, so the total bits for transmission is 4*6*2*32 bits.

According to the energy model for communications in [30], we can calculate the energy consumption for communication between Camera i and Camera j. The average energy consumption for communication to exploit correlation is illustrated in Fig. 10. Fig.10(a) is a comparison of energy consumption for both schemes, and Fig.10(b) shows the energy consumption per node for the proposed scheme. For both schemes, the energy consumption per node increases as the distance between the two nodes increases. But the proposed correlation model requires much less energy for communication than the feature extraction scheme.

It is commonly believed that communication is the most energy consuming operation for sensors, which requires much more energy than processing, however, due to the complexity of processing algorithms for visual information, the energy consumption for processing visual information is not negligible. The feature extraction algorithm [43] depends on wavelet transform that makes the energy consumption for computation comparable to communication energy dissipation [39]. Moreover, as image processing schemes are usually implemented in the unit of pixels, the energy consumption for processing is proportional to the size/resolution of the observed image. When the resolution increases, the energy consumption of the image processing based scheme will increase accordingly. In contrast, the computation process for the proposed model is very simple and straightforward, and the energy consumption for computation will not be influenced by image resolutions.

We have evaluated the validity as well as the costs of the proposed spatial correlation coefficient. From the above experimental results, we can conclude that the proposed correlation coefficient can effectively model the correlation characteristics of visual information through low computation and communication costs.

2.5.2 Joint Effect of Multiple Cameras

In this section, we present a set of simulations to evaluate the joint effect of multiple cameras and the correlation-based camera selection algorithm. In a field of 500m*500m, we set an area of interest that is located in the center of the field and has a radius of 10 meters. We randomly deploy N cameras that can observe this area of interest. Let M be the number of cameras to be selected by the sink to transmit their observed images. Suppose each camera obtains one image about the area of interest. Let $\{X_1, X_2, ..., X_N\}$ denote the images observed by these N cameras, and let $\{X_{i1}, X_{i2}, ..., X_{iM}\}$ denote the images observed by the M selected cameras.

Without loss of generality, we assume that the entropy of a single image is a constant value, denoted as $H(X_i) = H(\cdot)(i = 1, 2, ..., N)$. For these N cameras, we can obtain a correlation matrix $(\rho_{ij})_{N*N}$ as introduced in Section 2.3. So the joint entropy of $H(X_1, X_2, ..., X_N)$ and $H(X_{i1}, X_{i2}, ..., X_{iM})$ can be estimated using *Algorithm 1.*

We compare the following two camera selection schemes:

- *Random selection.* Randomly select M cameras out of the N cameras. For each M, repeat the experiment for 50 times. Compute the joint entropy at each time, and take the average value of the 50 trials as the final joint entropy.

- *Correlation-based selection.*This is the proposed method described in *Algorithm 2.* It makes use of correlation by selecting a group of M cameras that are least correlated with each other, so that the amount of information from the selected cameras can be maximized.

In our first experiment, we randomly deploy 10 cameras in the field ($N = 10$), and let M change from 2 to 10. The results of both schemes are shown in Fig. 11. The value of joint entropy increases as the number of nodes increases, which indicates that if more cameras transmit their observed images to the sink, more information

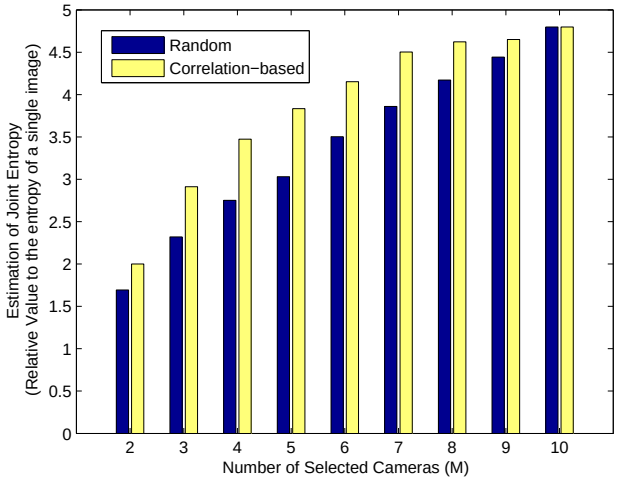

Figure 11: Estimation of joint entropy.

can be gained about the area of interest at the sink. When $M = 10$, all the cameras are selected to transmit their obsereved images, so both schemes produce the same results. But for $M = 2$ to 9, the correlation-based algorithm always results in larger joint entropy than the random selection of cameras.

According to the numerical results, when the number of selected cameras are the same for these two schemes, the correlation-based algorithm can increase the joint entropy by $0.5466 * H(\cdot)$ in average (increase by 18.37% in average compared to the random selection algorithm). It should be noted that the values of joint entropy in our simulation are expressed as relative values to the entropy of a single image, $H(\cdot)$. We find in our experiments that a typical value of $H(\cdot)$ is 5-6 bits/pixel for images of 8-bits depth, so the correlation-based scheme can result in about 3 bits/pixel increase in joint entropy than the random selection scheme.

Next we introduce more simulations to evaluate the distortion performance of both schemes. We implement both schemes for three different network topologies, where the total number of cameras, N, equals to 6, 10, and 15, respectively. Fig. 12 plots the distortion performance of both schemes. The distortion decreases as the number of selected nodes increases. For the same number of selected cameras (M),

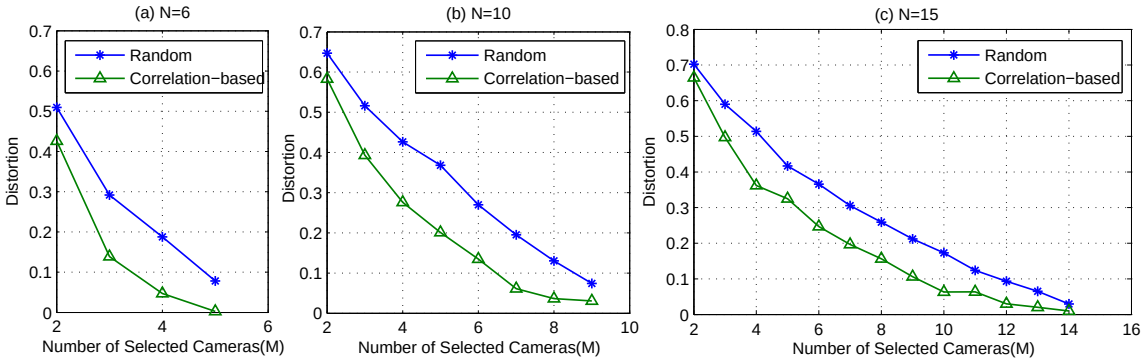

Figure 12: Distortion function.

the proposed correlation-based scheme results in lower distortion compared to the random selection scheme.

From another perspective, if a certain distortion bound is required at the sink, we may need fewer cameras to transmit their information using the correlation-based selection scheme. For example, in Fig. 12(b), a total number of 10 cameras are deployed to observe an area of interest. If the sink wants to obtain 80% of the total information, the maximum distortion is 0.2. As shown in Fig. 12(b), 7 cameras are needed on average when cameras are randomly selected, but only 5 cameras are needed when the correlation-based selection scheme is used. Therefore, given a distortion bound at the sink, the correlation-based selection scheme requires fewer cameras to report to the sink than the random selection scheme.

CHAPTER III

A COLLABORATIVE IMAGE COMPRESSION FRAMEWORK USING CLUSTERED SOURCE CODING

3.1 Introduction

Multimedia source coding [69, 51] is a common approach to remove the redundancy of visual information. However, the resource constraints of the sensor nodes bring new challenges when applying source coding *globally* in the entire network. The conventional video coding standards, such as MPEG/H.26x [69], can achieve high compression performance. However, they require extensive computation at the encoder, which places heavy burden on the resource-constrained sensor nodes. In [70] and [39], energy-efficient image compression is achieved by distributing the workload of compressing an image over several adjacent sensor nodes. Although promising for compressing the images generated by a single node, these solutions do not explore the correlation of the observed images among adjacent sensors. In contrast, distributed source coding, such as Slepian-Wolf Coding [57], only requires low-complexity encoding and leaves the intensive computations at the decoder. However, this coding strategy requires each sensor node to have the knowledge of global correlation structure, which would incur significant additional costs. For these reasons, multimedia source coding is infeasible to be applied *globally* in a large-scale network.

The clustering strategy has been proved to be an effective way to improve network scalability and energy efficiency for sensor networks [31, 8]. This strategy uses the hierarchical concept where the entire network is divided into regions. In many existing algorithms, the metrics for clustering are distance between nodes or node residual energy [4]. However, in this work, we aim to construct clusters based on the potential

coding gains so as to minimize the redundancy of network traffic. We divide the entire network into different regions. Each region corresponds to a cluster, in which a group of camera sensors collaboratively perform data compression, according to different coding algorithms. In the case of conventional coding standards, a powerful cluster head, such as the GARCIA robotic platform [5], can be placed within each cluster to serve as a single encoder, which has all correlated multimedia streams as inputs, thereby avoiding the computationally intensive operations draining the limited sensor energy store. In contrast to the conventional coding schemes that require centralized realization, distributed source coding allows each sensor to encode its own data separately, assuming a priori knowledge of local correlation structure in its own cluster [57]. Since each cluster only covers a limited number of nodes, it is feasible to acquire the correlation within a cluster without incurring much extra cost. Therefore, the clustered coding strategy paves the way for the practical application of multimedia source coding in large-scale WMSNs.

In this chapter, we propose an information theoretic data compression framework that maximizes the overall compression gain of the visual information retrieved from a WMSN. This framework consists of two components: (i) compression efficiency prediction, and (ii) coding hierarchy construction. Both components are independent of the specific coding algorithms and images types, thus providing a generic architecture that allows users to freely customize the WMSN applications based on them. The compression efficiency prediction aims to estimate the compression gain from joint encoding of multiple cameras before the actual images are captured. To achieve this, an entropy-based divergence measure (EDM) is proposed, which only takes the camera settings as inputs without requiring the statistics of real images. In the EDM, the overlapping pattern of the FoVs of multiple cameras is first identified. Then, the correlation degree among the observations from cameras with overlapped FoVs is obtained through a spatial correlation model. Based on the correlation characteristics,

a dependency graph based algorithm is designed to estimate the joint entropy of multiple cameras. This joint entropy effectively predicts the compression performance for joint encoding of multiple cameras.

Using the results from EDM, the next problem is how to establish a compression-oriented coding hierarchy, which can achieve a substantial compression gain and decoding reliability. This problem can be further formulated as an optimal coding clustering (OCC) problem, which we define as: find a set of coding clusters with the minimum total entropy, such that each camera node is covered by at least two different clusters. The minimization of total entropy guarantees that the global compression gain is maximized, while the coverage requirement ensures that the impact of cluster failures on the decoding reliability is mitigated. We prove that the OCC problem is NP-hard. As a heuristic solution, a fully distributed protocol, called distributed multi-cluster coding protocol (DMCP), is presented to provide a $\ln \Delta$ approximation to the optimal solution, where Δ is the maximum node degree in the network. Moreover, it is shown that $\ln \Delta$ is the best achievable approximation ratio for the OCC problem.

The rest of this chapter is organized as follows. Section 3.2 mathematically formulates the problems in the proposed data compression framework. In Section 3.3, the EDM algorithm is introduced to provide a valid assessment of joint coding performance of multiple cameras. The DMCP for establishing the efficient and robust coding hierarchy is proposed in Section 3.4. The performance of this framework is evaluated in Section 3.5

3.2 Problem Formulation

3.2.1 Spatial Correlation of Visual Information

In a WMSN, multiple camera sensors are deployed to provide multiple views, multiple resolutions, and enhanced observations of the environment. As shown in Fig. 13,

multiple cameras are deployed in a field of interest, and the cameras' field of views (FoVs) overlap with each other. A camera can only observe the objects within its FoV. And the sensing process of a camera is characterized by projection from a 3-D scene to a 2-D image. The observed images from cameras with overlapped FoVs are correlated with each other. We define the correlation of observed images caused by overlapped FoVs as *spatial correlation* in our context. The spatial correlation of the observed images further leads to data redundancy in WMSNs.

For two arbitrary camera sensors C_j and C_k with FoVs $\mathcal{A}_j$ and $\mathcal{A}_k$, suppose at a same time, their observed images are X_j and X_k, respectively. X_j and X_k are correlated when $\mathcal{A}_i$ and $\mathcal{A}_j$ overlap with each other. We introduce a spatial correlation coefficient $\rho_{j,k}$ to quantify the degree of correlation between X_j and X_k. This coefficient will be used as an important parameter in the following problems.

3.2.2 Clustered Source Coding

To remove the redundancy for correlated cameras, a group of camera sensors can form a cluster to collaboratively compress their data. Consider a cluster consisting of a cluster head (CH) and N camera sensors, where each sensor i produces image X_i, which is encoded with rate R_i. According to basic coding theorems, we have the following observation:

Observation 1. *The total coding rate of all nodes within a cluster is lower bounded by the joint entropy $H(X_1, X_2, \ldots, X_N)$ no matter centralized or distributed source coding is applied.*

For centralized source coding, each member in a cluster sends its raw or preprocessed data to the CH, while the CH acts as a single encoder that takes all collected data as inputs. According to Shannon's source coding theorem [12], each cluster can generate a total rate lower-bounded by the joint entropy $H(X_1, X_2, \ldots, X_N)$, which

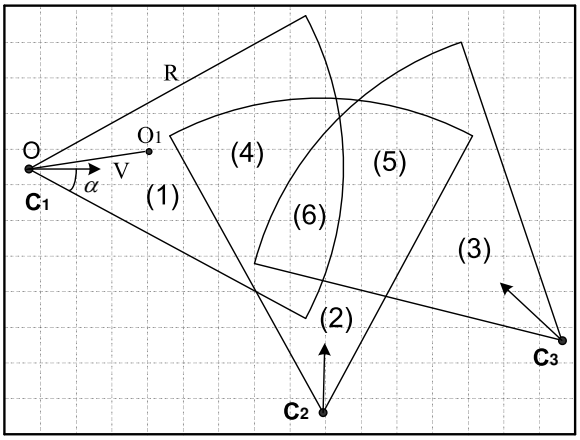

Figure 13: Field of views of multiple cameras.

is given by

$$\sum_{i=1}^{N} R_i \geq H(X_1, X_2, \cdots, X_N) \tag{22}$$

where the equality holds when an optimal encoder is used.

For distributed source coding (DSC), each node in a cluster separately encodes its own data, and the CH only acts as a relay node to forward the received data to data sink, where the compressed data are jointly decoded. In this case, Slepian-Wolf coding theorem [57] provides a conceptual basis for DSC and establishes the rate region for the rate vector $(R_1, R_2, ...R_N)$:

$$\sum_{i \in U} R_i \geq H(X(U)|X(U^c)) \quad \forall U \subseteq \{1, 2, \cdots, N\} \tag{23}$$

where $X(U) = \{X_j | j \in U\}$ and U^c is the complementary set of U.

Surprisingly, Slepian-Wolf coding theorem (23) indicates that the sum of rates, $\sum_{i=1}^{N} R_i$, can achieve the joint entropy $H(X_1, X_2, \cdots, X_N)$, just as for joint encoding the sources $(X_1, X_2, \cdots, X_N)$, despite separate encoders for them. Therefore, a cluster with N nodes can be optimally encoded with $H(X_1, X_2, \cdots, X_N)$ bits no matter centralized or distributed source coding is applied.

3.2.3 Multi-camera Entropy Estimation Problem

Joint entropy serves as a lower bound of the overall coding rate of multiple sources for both centralized and distributed source coding. If the joint entropy for a cluster of cameras can be estimated, we will be able to predict the performance of joint coding within the cluster. However, to estimate the joint entropy of visual information from multiple cameras is a challenging task. Because of the intrinsic complexity of visual information, it is difficult to model the dependency characteristics of visual sources, and moreover, it usually requires expensive computation and communication costs.

Our objective is to estimate the joint entropy of multiple cameras in WMSNs through low computation and communication costs. Given a cluster of cameras with observations $X_1, X_2, \cdots, X_N$, the joint entropy $H(X_1, X_2, \cdots, X_N)$ will be described as a function of the individual entropy ($H(X_i)$) and field of view ($\mathcal{A}_i$) of each camera, and the correlation coefficients between any two cameras ($\rho_{j,k}$) in the cluster.

3.2.4 Optimal Coding Clustering Problem

Since joint entropy provides a benchmark on the compression gain from joint encoding of multiple sources, we can utilize a similar entropy-based concept, called *cluster entropy*, to measure the collaborative compression gain within the scope of a single *coding cluster*. The target of optimal coding clustering can then be correspondingly interpreted as to select a set of *coding clusters* according to their *cluster entropies* such that total entropy of the entire network is minimized. We describe two definitions involved in the discussion above.

Definition 1. *A* coding cluster *is a finite set comprising a camera sensor and all sensors within its transmission range.*

Definition 2. *For each coding cluster A, its* cluster entropy $H(A)$ *is equal to the joint entropy of all cameras in A.*

Now, the Optimal Coding Clustering (OCC) problem can be formally stated as: given a network consisting of a finite set of camera sensors $E = e_1, e_2, ...e_n$ and a set of n subsets of E, $\mathcal{S} = \{S_1, S_2, ...S_n\}$, where each set S_i corresponds a coding cluster with its entropy $H(S_i)$, the goal is to find a collection C from $\mathcal{S}$ of minimum total entropy $\sum_{S_i \in C} H(S_i)$, such that each element e_i is covered by at least two sets in C.

The minimization of total entropy guarantees that the maximum global compression gain is achieved, while the coverage requirement ensures that the visual information encoded by each camera has more chance to be successfully delivered to, and properly decoded at data sink.

3.3 Joint Entropy Estimation

In this section, we propose a novel Entropy-based Divergence Measure (EDM) scheme to estimate the joint entropy of observations from multiple cameras. This scheme only takes cameras' settings as inputs without requiring the knowledge of specific applications, thereby providing a generic framework for prior evaluation of compression under different coding solutions. Moreover, it induces little communication costs since camera nodes only need to exchange their settings via short messages among their 1-hop neighbors, and low complexity computations are required for joint entropy estimation. The EDM scheme consists of the following two components.

1. *Area division for FoVs.* Given a group of cameras, their FoVs are divided into several regions, such that each region is covered by the same set of cameras.

2. *Joint entropy estimation for regions.* For each region, a dependency graph is constructed based on the correlation among the cameras. The joint entropy of the region is then estimated by traversing the dependency graph. Finally, the total joint entropy for the group of cameras is the sum of the entropies of all the regions.

3.3.1 Area Division for Overlapped Field of Views

A camera is a directional sensor with limited sensing range. It can only observe the objects within its field of view (FoV). If a camera sensor is deployed on a ground plane, we can use a simplified 2-D FoV model [42] that models the shape of a camera's FoV as a sector. As shown in the left part of Fig. 13, a camera's FoV is determined by four parameters: O, R, $\vec{V}$, and α, where O is the location of the center of the camera, R is the sensing radius, $\vec{V}$ is the sensing direction (the center line of sight of the camera's FoV), and α is the offset angle between the sensing direction and a radius of the sector. An arbitrary point O_1 is in the FoV of the camera if it satisfies

$$\begin{cases} |O\vec{O}_1| \leq R \\ \theta \leq \alpha, \end{cases} \tag{24}$$

where θ is the angle between $O\vec{O}_1$ and $\vec{V}$.

Another key parameter for a camera's sensing process (3-D to 2-D projection) is the camera's focal length (f). Both the FoV parameters and the focal length could be estimated through calibration methods for WMSNs, e.g., [18].

We consider the case when N cameras $(C_1, C_2, \cdots, C_N)$ are deployed on the ground plane and all the cameras are homogeneous, i.e., they have the same focal lengths (f), sensing radiuses (R), and offset angles (α). Denote the FoV of an individual camera C_i by $\mathcal{A}_i(O_i,R_i,\vec{V}_i,\alpha_i)$, and the overall FoV for these cameras by $\mathcal{A}$ ($\mathcal{A} = \{\mathcal{A}_1, \cdots, \mathcal{A}_N\}$). The goal of area division is to divide $\mathcal{A}$ into several regions $(P_1, P_2, \cdots, P_M)$, such that each region belongs to the FoVs of the same set of cameras. As shown in Fig. 13, the FoVs of the three cameras are divided into six different regions.

We introduce a grid-based approach to divide the overall FoV $\mathcal{A}$ into regions. As shown in Fig. 13, the overall FoV $\mathcal{A}$ is divided into small grids $(G(k), k = 1, \cdots, K)$. And then we can check if a grid $G(k)$ is in a camera's FoV $(\mathcal{A}_n, n = 1, \cdots, N)$ as

follows: we find the center point of the grid, and using the condition in (24), we can tell if it is in the camera's FoV; if this center point is in the camera's FoV, we regard that this grid is in the camera's FoV. (This approximation is valid as long as the size of the grid is much smaller than the size of the FoV.) After traversing all the grids in the overall FoV, regions could be formed by grouping the grids that belong to the same set of cameras.

3.3.2 Estimating the Joint Entropy of a Region

In this section, we introduce an algorithm to estimate the joint entropy of a region. Denote the cameras that can observe region P_i by $(C_1, \cdots, C_n)$. For the kth camera C_k, denote its observed visual information by X_k, and denote its observation about this region by $X_k(P_i)$. The amount of information of the region P_i is the joint entropy of the observations about this region from the cameras $(C_1, \cdots, C_n)$, given by

$$H(P_i) = H(X_1(P_i), \cdots, X_n(P_i)). \tag{25}$$

Since there is no unified probability model for images, and estimating the joint probability distribution of multiple sources requires large bulk of computation, it is difficult to calculate joint entropy in resource-constrained WMSNs. (See Chapter 2 and [15] for details.) In this chapter, we introduce a novel approach to estimate joint entropy based on the spatial correlation model [15] in Chapter 2. Our solution consists of three steps:

1. Estimate the individual entropy $H(X_k(P_i))$ in (25);
2. Study the correlation characteristics among the individual observations using the correlation model in Chapter 2, which can also be found in [15];
3. With the results from the above two steps, apply a dependency graph based algorithm to estimate $H(P_i)$ in (25).

We explain these steps in details in the following paragraphs.

3.3.2.1 Individual Entropy Estimation

For an arbitrary camera C_k, the entropy of its observed image X_k is $H(X_k)$. The entropy of the observation X_k about the region P_i, $H(X_k(P_i))$, can be estimated as

$$H(X_k(P_i)) \approx \frac{S(P_i)}{S(\mathcal{A}_k)} H(X_k) \tag{26}$$

where $S(P_i)$ is the area of P_i and $S(\mathcal{A}_k)$ is the area of the FoV. The entropy $H(X_k)$ is the total amount of information of X_k, which is provided by the projections of all the 3-D points in the FoV. As there is no prior knowledge about where a camera is deployed or what type of scene is observed, it is assumed that when all the 3-D points in the FoV are projected on the camera's image plane, each point provides the same amount of information. Considering that the cameras are deployed on a ground plane and a 2-D FoV model is used, the amount of information that camera C_k contributes to P_i is approximately proportional to the area of P_i, so we use the ratio $\frac{S(P_i)}{S(\mathcal{A}_k)}$ to estimate $H(X_k(P_i))$ in (26).

This assumption works well when there are no large obstacles in a camera's FoV. A camera's FoV might be reduced in case of obstacles, therefore, when implementing the proposed algorithm in practical applications, we might need to update the camera's FoV model to reflect the effect of obstacles.

There are many different models for images, and different values of entropy may be obtained for the same image sources. In our algorithm, we avoid calculating the exact values of entropy of images. As we consider the case that all the camera sensors in a WMSN are homogeneous, without loss of generality, we assume that the entropies of the single observed images are the same, denoted by $H(X_i) = H(\cdot)(i = 1, \cdots, N)$. All the joint entropy terms in our algorithm will be expressed as relative values of $H(\cdot)$.

3.3.2.2 Spatial Correlation Motivated Entropy Estimation

In our earlier work, we proposed a novel spatial correlation model for visual information in WMSNs [15]. Given an area of interest and two cameras that can observe it, a spatial correlation coefficient was derived to quantify the degree of correlation between the two cameras. For example, if we take region P_4 in Fig. 13 as the area of interest, both camera C_1 and camera C_2 can observe it, with observations $X_1(P_4)$ and $X_2(P_4)$. We pick some reference vectors in region P_4, and calculate the projections of these reference vectors in C_1 and C_2 using the projection model of cameras. By studying the correlation between the projected reference vectors on the two cameras, the spatial correlation coefficient $\rho_{1,2}$ can be calculated.

In general, for cameras C_j and C_k that can observe region P_i, with P_i as the area of interest, a spatial correlation coefficient between the observations of P_i at C_j and C_k was derived as a function as follows:

$$\rho_{j,k} = f(O_j, \vec{V_j}, O_k, \vec{V_k}, P_i) \tag{27}$$

where O_j and O_k are the two cameras' locations, $\vec{V_j}$ and $\vec{V_k}$ are their sensing directions, The spatial correlation coefficient was designed as a normalized symmetric metric, i.e., it satisfies $\rho_{j,k} = \rho_{k,j}$ and $0 \leq \rho_{j,k} \leq 1$.

More importantly, the spatial correlation coefficient was related to the estimation of joint entropy in [15]. We briefly introduce the relevant results here.

To evaluate the dependency between two visual sources, an *entropy correlation coefficient (ECC)* was introduced in [50]. The ECC for two visual sources A and B was defined as

$$ECC = \frac{2I(A;B)}{H(A) + H(B)}. \tag{28}$$

Moreover, the joint entropy $H(A, B)$ can be given as

$$H(A, B) = (1 - \frac{1}{2}ECC)(H(A) + H(B)). \tag{29}$$

By definition, the joint probability distribution of the two sources is needed to estimate the joint entropy. Due to the complexity of image contents and the difficulty in image modeling, it is difficult to get an accurate estimation of the joint probability distribution [50]. Besides, camera sensors in a WMSN must exchange their observed images to estimate the joint probability distribution, which introduces a lot of communication burden in the network. In [15], it was found that the spatial correlation coefficient (27) had the same intrinsic meaning as ECC: both ranging from 0 to 1 and denoting the degree of correlation between two sources, while the spatial correlation coefficient could be obtained through low computation and communication costs. Therefore, the ECC term in (29) was replaced by the spatial correlation coefficient.

Consequently, for cameras C_j and C_k that can observe region P_i, the joint entropy of the observations of P_i at C_j and C_k was estimated as

$$H(X_j(P_i), X_k(P_i)) \approx (1 - \frac{1}{2}\rho_{j,k})(H(X_j(P_i)) + H(X_k(P_i))) \tag{30}$$

where $X_j(P_i)$ is the observation of P_i at camera C_j, and $X_k(P_i)$ is the observation of P_i at camera C_k. This equation indicates that the amount of information gained from the observations of two cameras depends on the correlation between them. The more the two observations are correlated, the less joint entropy can be gained from them together.

From (19) we can obtain the conditional entropy as follows:

$$\begin{aligned} H(X_j(P_i)|X_k(P_i)) &= H(X_j(P_i), X_k(P_i)) - H(X_k(P_i)) \\ &\approx (1 - \frac{\rho_{j,k}}{2})H((X_j(P_i)) - \frac{\rho_{j,k}}{2}H(X_k(P_i)) \end{aligned} \tag{31}$$

where $H(X_j(P_i)|X_k(P_i))$ is the entropy of $X_j(P_i)$ with the knowledge of $X_k(P_i)$.

3.3.2.3 Dependency Graph Based Joint Entropy Estimation

Based on the correlation coefficient (27) and the conditional entropy term (31), we propose a dependency graph based algorithm to estimate the joint entropy of a region.

We study a two cameras' case first. Suppose there are only two cameras (C_1 and C_2) in a region P_i. We can depict their relationship using a dependency graph: $C_2 \rightarrow C_1$. The joint entropy of the observations from C_1 and C_2 can be calculated by traversing the dependency graph. The source node C_2 contributes the entropy of its observations, $H(X_2(P_i))$, and the node C_1 contributes the conditional entropy with respect to C_2, $H(X_1(P_i)|X_2(P_i))$, so the joint entropy is calculated by adding these two terms: $H(X_1(P_i), X_2(P_i)) = H(X_2(P_i)) + H(X_1(P_i)|X_2(P_i))$. The dependency graph can also be constructed as $C_1 \rightarrow C_2$, from which we can get the same result of joint entropy.

The two cameras' case can be extended to estimate the joint entropy of more than two cameras. Generally, for an arbitrary number of cameras, we construct a dependency graph to describe the dependency characteristics among them. Denote the dependency graph by $G(V, E)$, where V is a collection of cameras, and E is a collection of directed edges that stand for dependencies. Joint entropy of the region is calculated by traversing all the nodes in the graph along the directed edges. The detailed steps are described in *Algorithm* 3.

For a group of cameras $(C_1, C_2, \cdots, C_n)$ that can observe the region P_i, we can obtain a correlation matrix $(\rho_{j,k})_{n*n}$ based on (27). To simplify the problem, we assume limited number of dependencies: each camera is dependent on the camera that is most correlated with it. For example, if camera C_j is most correlated with camera C_k, we say that C_j is dependent on C_k, and we can construct a directed edge starting from C_k and ending at C_j: $C_k \rightarrow C_j$. C_j is said to be a direct successor of C_k, and C_k is a direct predecessor of C_j.

The dependency graph is designed to be a directed acyclic graph with the following constraints: a camera is either a source node (i.e., a node that has no predecessors), or a direct successor of one of the other cameras; a dependency graph may have several source nodes, but each node can have at most one direct predecessor; and

there should be no loops in the graph, e.g., $C_k \rightarrow C_j$ and $C_j \rightarrow C_k$ cannot exit in the same graph. These properties could be guaranteed through the procedure of constructing the dependency graph (lines 5-12 in *Algorithm* 3). For each node C_k, if another node C_j is most correlated with it, i.e., $neighbor(C_j) = k$, the algorithm adds $C_k \rightarrow C_j$ into the graph only when two conditions are met: i) C_j has no predecessors, and ii) C_j is not a predecessor of C_k (line 7 in *Algorithm* 3). The first condition guarantees that each node can have at most one direct predecessor, and the second one guarantees that there are no loops in the graph.

Given a dependency graph with the above features, the joint entropy is estimated by traversing all the nodes in the graph and adding the entropies of the nodes together, which corresponds to lines 13-19 in *Algorithm* 3. A source node contributes its individual entropy to the joint entropy, while a non-source node contributes its conditional entropy with respect to its direct predecessor to the joint entropy.

Algorithm 3 Dependency Graph Based Entropy Estimation

1: P_i: $\{C_1, C_2, ..., C_n\}$ with correlation matrix $(\rho_{j,k})_{n*n}$.
2: **for** $j = 1$ to n **do**
3: $neighbor(C_j) = arg \max_{k \neq j}(\rho_{j,k})$;
4: **end for**
5: **for** $k = 1$ to n **do**
6: **for** $j = 1$ to n and $j \neq k$ **do**
7: **if** $neighbor(C_j) =$ k and C_j has no predecessors and C_j is not a predecessor of C_k **then**
8: Add $C_k \rightarrow C_j$ into the dependency graph;
9: $Predecessor(C_j) = C_k$;
10: **end if**
11: **end for**
12: **end for**
13: **for** $j = 1$ to n **do**
14: **if** C_j has no predecessor **then**
15: Add $H(X_j(P_i))$ to $H(P_i)$;
16: **else if** $Predecessor(C_j) = C_k$ **then**
17: Add $H(X_j(P_i)|X_k(P_i))$ (31) to $H(P_i)$;
18: **end if**
19: **end for**
20: **return** $H(P_i)$.

Since the FoVs for a group of cameras are divided into several independent regions, the total joint entropy is the sum of the entropies of all the regions. For a group of cameras with observations $(X_1, \ldots, X_N)$, with their FoVs divided into regions $(P_1, \ldots, P_M)$, the total joint entropy is given by

$$H(X_1, \cdots, X_N) = H(P_1) + \cdots + H(P_M) \tag{32}$$

where $H(P_i)(i = 1, \cdots, M)$ is obtained by *Algorithm* 3.

To provide an overview of the EDM algorithm, we illustrate the steps for estimating the joint entropy of the three cameras in Fig. 13. The FoVs of the three cameras are divided into six regions. We take the sixth region (P_6) as an example. All the three cameras (C_1, C_2, C_3) can observe P_6. Suppose we find from the geometry of the cameras' FoVs that $S(P_6) = 0.1S(\mathcal{A})$. The entropy of a single image is $H(\cdot)$, by applying (26), the individual entropies about this region are $H(X_i(P_6)) = 0.1H(\cdot)(i = 1, 2, 3)$. Furthermore, from (27) we can obtain a correlation matrix for P_6:

$$(\rho_{jk})_{3*3} = \begin{pmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix}.$$

By applying *Algorithm* 3 on the correlation matrix, we can obtain a dependency graph as $C_3 \rightarrow C_2 \rightarrow C_1$. Therefore, the joint entropy of region P_6 is $H(P_6) = H(X_3(P_6)) + H(X_2(P_6)|X_3(P_6)) + H(X_1(P_6)|X_2(P_6))$, where the conditional entropies can be calculated from (31). For example, $H(X_2(P_6)|X_3(P_6)) = (1 - \frac{\rho_{2,3}}{2}) \cdot H(X_2(P_6)) - \frac{\rho_{2,3}}{2} \cdot H(X_3(P_6)) = 0.05H(\cdot)$. In the same way, we can obtain $H(X_1(P_6)|X_2(P_6)) = 0.09H(\cdot)$. Thus, the joint entropy of P_6 is $H(P_6) = 0.24H(\cdot)$. After the entropy of each region is calculated, the joint entropy of the three cameras is calculated by $H(X_1, X_2, X_3) = H(P_1) + \cdots + H(P_6)$.

The entire EDM algorithm can be run at each sensor node. To estimate joint entropy, a node just need to acquire the FoV parameters, locations, and sensing

directions of its neighbors. Therefore, it does not require expensive communication costs in the network. The estimated joint entropy will serve as a criteria for the DMCP protocol in the following section.

3.4 Data Compression using Clustered Source Coding

After a WMSN is deployed in a field, we would like to select a set of coding clusters to cover the entire network with maximum compression ratio. Due to the distributed manner of WMSNs and the changing environment, a centralized algorithm is not suitable for use here. The coding cluster selection should only depend on local information. In this section, we first formulate the optimal coding clustering (OCC) problem as an integer program, and shows that the OCC problem is NP hard. Accordingly, we propose a distributed multi-cluster coding protocol (DMCP), which is shown to achieve an approximation guarantee of $\ln \Delta$, where Δ is the maximum node degree in the network.

3.4.1 Integer Program Formulation of OCC Problem

To formulate the OCC problem as an integer program, we assign a variable x_S for each set $S \in \mathcal{S}$, which is allowed 0/1 values. This variable will be set to 1 iff set S is selected for the coding hierarchy. The objective function is the sum of the entropy values of all selected coding clusters. The constraint is that for each node $e \in E$ we want that at least two of the clusters containing it are selected.

$$\begin{aligned} \text{MIN} \quad & \sum_{S \in \mathcal{S}} H(S) x_S \\ s.t \quad & \sum_{S: e \in S} x_S \geq 2, \quad e \in E \\ & x_S \in \{0, 1\}, \quad S \in \mathcal{S} \end{aligned} \tag{33}$$

If we treat $H(S)$ as the cost $c(S)$ associated with each coding cluster $S \in \mathcal{S}$ and let the second constraint be coverage requirement for each node $e \in E$, the OCC

problem can be reduced to the constrained set multicover (CSMC) problem. The CSMC problem is NP hard and the greedy algorithm is essentially the best one can hope for [53]. In other words, the approximation ratio $\ln \Delta$ achieved by the greedy algorithm is best one for CSMC problem. Therefore, the greedy strategy applies naturally to our OCC problem: let us say that the node e is uncovered if it occurs in fewer than 2 of the selected coding clusters. In each iteration, the algorithm selects, from the currently unselected clusters, the most compression-efficient cluster, where the compression efficiency of a cluster is defined to be the average entropy of the uncovered nodes it covers. The algorithm terminates when there are no more uncovered nodes, e.g., each node has been included by two different clusters. The pseudo-code of the above procedures is described in *Algorithm* 4.

Algorithm 4 Greedy Coding Cluster Selection Algorithm

$C \leftarrow \emptyset$, and $E' \leftarrow E$
For each $S \in \mathcal{S}$, $x(S) \leftarrow 0$.
while $E' \neq \emptyset$ **do**
  $s \leftarrow \arg\min_{X \in \mathcal{S}} H(X)/|X \cap E|$, and $x(s) \leftarrow 1$.
  $C \leftarrow C \cup s$, and $\mathcal{S} \leftarrow \mathcal{S} \setminus s$.
  $E' \leftarrow E' \setminus \{e \in s \cap E : \sum_{S:e \in S} x(S) \geq 2, S \in C\}$.
end while

The greedy algorithm for the OCC problem can be computed in $O(n)$ rounds if a central controller (e.g., data sink) provides the full information of the network topology along with the detailed settings (e.g., sensing direction, sensing offset angle, and sensing range) for each camera. However, in a large-scale distributed network like WMSN, the centralized operations have limited flexibility and scalability. Moreover, the energy constraint of sensor nodes prohibits network-wide information exchange. Next, we will propose a distributed protocol that only needs local information exchange to achieve global compression optimization.

3.4.2 Distributed Multi-Cluster Coding Protocol

After a WMSN is initially deployed, each camera node leads its neighbors to constitute a candidate coding cluster. At this time, each sensor node could be in one of the following four states: *black*, *grey*, *half grey*, and *white*. We call sensor nodes *black* if they are selected as the CH (CH) locaters. The CH locaters will not serve as the normal CHs but indicate the coordinates at which the future mobile or fixed CHs should be placed. We call the nodes *grey* if they are covered by at least two *black* nodes, and *half grey* if they are covered by exactly 1 *black* node. A node stays in the *white* state if there exists no *black* node within its 1-hop range. The *half grey* nodes and *white* nodes are collectively referred to as *uncovered* nodes. We now describe several useful definitions.

Definition 3. *The neighbor set of a node is a set consisting of the node itself and all nodes in its 1-hop range.*

Definition 4. *The serving set of a node is a set comprising the uncovered nodes that are residing in its 1-hop range.*

Definition 5. *The coding effectiveness of a node is the average entropy of all nodes in its serving set.*

Definition 6. *The CH counter of a node records the current number of the black nodes among its 1-hop neighbors.*

Now, the proposed DMCP establishes a clustered coding hierarchy as follows. Initially (lines 1 - 5 in *Algorithm* 5), no *black* nodes exist in the network. Thus, every node is *uncovered*. Nodes in the *uncovered* state send out their camera settings to their neighboring nodes. After receiving the setting information, an *uncovered* node discovers its *serving set* and calculates its *cluster entropy*. Based on these information, an *uncovered* node evaluates its *coding effectiveness*, which is sent out along with the node state in an advertising (ADV) message to its 2-hop neighbors.

Algorithm 5 Distributed Multi-cluster Coding Protocol

1: $state(e) \in \{black, grey, half\ grey, white, uncovered\}$
2: $state(e) \leftarrow white$, send & receive $state(e)$ and camera settings
3: $N_e \leftarrow \{e' : state(e') = white\} \cup \{e\}$
4: {Discover *neighbor set* N_e}
5: $counter(e) = 0$ {Set CH counter}
6: **while** $state(e) = uncovered$ **do**
7: $U_e \leftarrow \{e' \in N_e : state(e') = uncovered\}$
8: {Calculate *serving set* U_e}
9: $EC_e \leftarrow H(N_e)/\,|U_e|$
10: {Calculate *coding effectiveness* EC_e }
11: **if** $EC_e = \min_{e' \in U_e}\{EC_{e'}\}$ **then**
12: $state(e) \leftarrow black$, and $counter = 1$
13: send $COVERAGE$ msg
14: **else**
15: wait until the selection of a new *black* node times out
16: **if** no $COVERAGE$ received **then**
17: $state(e)$ remains
18: **else if** $counter = 0$ **then**
19: $state(e) \leftarrow halfgrey$, and $counter(e) = 1$
20: **else if** $counter = 1$ **then**
21: $state(e) \leftarrow grey$, and $counter(e) = 2$
22: send & receive ADV msg containing $state(e)$
23: **end if**
24: **end if**
25: **end while**
26: $Process\_Grey\_Black()$

A node in the *uncovered* (e.g., *half grey* or *white*) state collects ADV messages and extracts the *coding effectiveness* values from its 2-hop neighbors. If the node itself is the most coding-effective node amongst its 2-hop neighbors, it becomes a *black* node and sends COVERAGE messages to other *uncovered* nodes within its 1-hop range (lines 11 - 13 in *Algorithm* 5). Otherwise, an *uncovered* node can encounter the following scenarios: 1) if no COVERAGE message is received within the predefined maximum duration of selecting a new *black* node, the node remains *uncovered*, recalculates its *coding effectiveness*, and sends out an ADV message (lines 16 - 17 in *Algorithm* 5). 2) If a COVERAGE message is received, and its *CH counter* is equal to zero, the node enters *half grey* state and increments its *CH counter* by 1 (lines 18 - 19 in *Algorithm* 5). 3) If a COVERAGE message is received, and its *CH counter* already reaches 1, the node becomes a *grey* node and sets the *CH counter* to 2 (lines 20 - 21 in *Algorithm* 5). For the last two cases, a ADV message containing the node state is sent out to its immediate neighbors.

For a *grey* node, if the CH counters of all its neighbors already reach 2, the node remains *grey* for the rest of cluster selection procedure and becomes a cluster member in the end. Otherwise, the node sends out an ADV message containing its *coding effectiveness* and collects ADV messages from all the *uncovered* nodes within its 2-hop range. If the node itself has the highest *coding effectiveness*, it enters *black* state and send out COVERAGE messages to its *uncovered* neighbors (lines 5 - 6 in *Algorithm* 6). Otherwise, if the maximum duration of generating a new *black* node passes, and there still exist *uncovered* nodes within its 1-hop range, the node remains *grey* (lines 8 - 10 in *Algorithm* 6). A *black* finally becomes a CH locater until the value of its CH counter reaches 2 on receiving a COVERAGE message (lines 13 - 17 in *Algorithm* 6).

The above procedures are performed by all nodes until each of them becomes either a CH locater or a cluster member. At the end, there is no *uncovered* node in

Algorithm 6 *Process_Grey_Black()*

```
1: U_e ← {e ∈ N_e : counter(e) < 2}
2: while |U_e| < |N_e| do
3:    if state(e) = grey then
4:       recalculate coding effectiveness EC_e
5:       if EC_e = min_{e'∈U_e}{EC_{e'}} then
6:          state(e) ← black, and send COVERAGE msg
7:       else
8:          wait until the new black selection times out
9:          if |U_e| < |N_e| then
10:            state(e) ← grey
11:         end if
12:      end if
13:   else if state(e) = black then
14:      wait until a COVERAGE is received
15:      counter(e) = 2
16:      send ADV msg containing state(e) and counter(e)
17:      Node e becomes a CH locator
18:   end if
19: end while
20: Node e becomes a cluster member
```

the network, and the established clustered coding hierarchy covers the entire network. The pseudo-code of the above procedures is described in *Algorithm* 5 and *Algorithm* 6.

3.4.3 Correctness and Complexity

Theorem 1. *If the minimum node degree in a network is 2, each node will be covered by at least two coding clusters when DMCP terminates.*

Proof. Assume when DMCP terminates, a node v does not belong to any coding cluster or it is only covered by one coding cluster. This implies that v stays in the *uncovered* state. Thus, the condition in line 6 of *Algorithm* 5 is satisfied. Since the minimum node degree of v is larger than 1, v has at least one neighbor, say u. If u has higher coding efficiency than v, then u becomes a *black* node and v is covered by the coding cluster led by u. Otherwise, v enters *black* state, and thus v is covered by the coding cluster led by itself. Both cases contradict the assumption that v does not

belong to any coding cluster. On the other hand, if v is only covered by one cluster, this means that either v itself or one of its neighbors is a *black* node. In this case, v's neighbor or v will become a CH. (i.e., the operations in lines 12 - 14 of *Algorithm* 5 or lines 5 - 6 of *Algorithm* 6 are executed). This implies that v is covered by two clusters, thus contradicting the assumption that v is only covered by one coding cluster. □

Theorem 2. *The DMCP protocol has a worst case processing time complexity of* $O(N^2)$ *per node per round, where* N *is the number of nodes in the network.*

Proof. In *Algorithm* 5 and 6, the computational operations include two parts: the estimation of the cluster entropy (line 9 of *Algorithm* 5 and line 4 of *Algorithm* 6) and the search of the minimum average entropy (line 12 of *Algorithm* 5 and line 5 of *Algorithm* 6). The first part is calculated by the EDM scheme presented in Section 3.3. As indicated in *Algorithm* 3, the EDM has a time complexity of $O(N^2)$. The second part is realized by binary tree sorting, which takes $O(Nlog(N))$ iterations. Thus, in each round, DMCP protocol has a worst case processing time complexity of $O(N^2)$ per node. □

Theorem 3. *The DMCP terminates in* $O(N)$ *rounds, where* N *is the number of nodes in the network.*

Proof. Given a network with N sensor nodes, there exist total N candidate coding clusters, each of which consists of a sensor node and its neighboring nodes. As indicated in *Algorithm* 5, in each round, at least one candidate coding cluster is selected as the final coding cluster. Thus, the DMCP takes a time in $O(N)$ rounds in the worst case. □

Theorem 4. *The DMCP protocol has a worst case message exchange complexity of* $O(1)$ *per node.*

Proof. During the execution of *Algorithm* 5, an *uncovered* (*white* or $halfgrey$) node is silent until it sends notification messages $COVERAGE$ to become *black* node or sends

the joint messages ADV to become *grey* node. The number of these $COVERAGE$ messages are strictly less than N, since at most N nodes will enter *black* state. In addition, *uncovered* nodes generate at most N ADV messages, since at least one node will decide to be a CH. Besides *uncovered* nodes, during the execution of *Algorithm* 6 *black* nodes and *grey* nodes also send out ADV and $COVERAGE$ messages. Specifically, *black* nodes broadcast at most N ADV messages to advertise their final status, and *grey* nodes generate at most N $COVERAGE$ messages to announce a status change from the *grey* to the *black*. Hence, the number of messages exchanged in the network is upper-bounded by $4N$, i.e., $O(N)$. □

Since the clustered coding hierarchy only needs to be constructed when the network is initially deployed in the field of interest. Thus, the linear time and message complexity of the proposed protocol has trivial impact on the network performance, compared with the significantly enhanced energy efficiency induced by the established coding hierarchy. In addition, to reduce the computational delay, the protocol parameters, such as the duration of each round, can be the properly adjusted. The proposed framework can be implemented on a variety of camera sensor platforms, which are equipped with a wide range of microprocessors including ARM7, AVR, and Atmel ATmega128L [6]. Generally, users are provided with the dedicated compilers for the specific camera sensor platforms. To facilitate the cross compiling on different hardware platforms, we believe that the gcc compiler collection [1] can be a good choice because it is available for most embedded platforms equipped with a variety of microcontrollers.

3.4.4 Approximation Ratio

Theorem 5. *The DMCP computes a* $\ln \Delta$ *approximation for the optimal coding clustering problem.*

Proof. According to DMCP, the *cluster entropy* of a node is only related to camera

settings of the nodes in its *neighbor set*, and the *neighbor set* is only determined by the local topology, the value of the *cluster entropy* will not change as the protocol proceeds. On the other hand, the cardinality of the *serving set*, which is equal to the number of its *uncovered* neighboring nodes, can be reduced as protocol proceeds since some *uncovered* neighboring nodes could be included by some other clusters. Thus, we conclude that the *coding effectiveness* of a non-*black* node can only be reduced if the cardinality of its serving set decreases.

Based on this conclusion, we can further show that the DMCP is equivalent to the centralized greedy algorithm. According to DMCP, a non-*black* node v with the highest *coding effectiveness* within its 2-hop neighborhood is eligible to become a *black* node. The selection of other non-*black* nodes outside v's 2-hop range as *black* nodes will not affect v's eligibility to enter the *black* state because the status change of the nodes outside v's 2-hop range can not reduce v's *serving set* cardinality, and according to the conclusion above, v's *coding effectiveness* remains the same. Therefore, the DMCP chooses v as a *black* node before any nodes within its 2-hop range. On the other hand, the centralized greedy algorithm always selects the most *compression efficient* cluster, and v leading its neighbors represents the most *compression efficient* cluster within its 2-hop range. Therefore, the centralized approach will select the cluster led by v as a final coding cluster as the algorithm proceeds. This means that the DMCP obtains the same result as the centralized algorithm, thus achieving the same $\ln \Delta$ approximation ratio as the centralized algorithm. □

As shown in Section 3.4.1, the OCC problem can be reduced to CSMC problem, for which $\ln \Delta$ is the best approximation ratio. Thus, we can conclude that no protocols perform better than the proposed DMCP in terms of application factor.

3.4.5 Intercluster Connectivity

After the DMCP is performed, the selected CH locaters send out messages to advertise their states and coordinates. Then, some more powerful multimedia nodes, such as the GARCIA robotic platform [5], can automatically move to or be manually placed at these locations, and act as normal CHs. Since the CHs are interconnected by multi-hop connections, the CHs should properly adjust their transmission range to maintain inter-cluster connectivity. We address this problem by proving the following theorem.

Theorem 6. *In a WMSN with the minimum node degree $\delta \geq 1$, i.e., there is no isolated node in the network, any two CHs are two hops away at most.*

Proof. Since every node has at least one neighbor, each cluster member belongs to at least two clusters after the DMCP is performed. This means that each cluster member has two different CHs within its 1-hop range. Suppose a CH v can reach the nearest CH w at least three hop away. Then, there exists a cluster member u of the CH v in the path between v and w. This implies that the CH w is at least 2-hops away from the cluster member u. Therefore, u can not be covered by the CH w, and there has to be another CH, say x, within 1-hop range of u to meet its coverage requirement. Now, the cluster member u has two CHs, v and x, within its immediate neighborhood. This means that v and x are at most 2 hops away from each other, which contradicts with the assumption that a CH v can reach the nearest CH at least three hops away. □

Therefore, in order to maintain inter-cluster connectivity, each CH only needs to adjust its transmission range to twice the 1-hop distance.

Figure 14: (a) Indoor scene "Tables". (b) Outdoor scene "Trees".

3.5 Performance Evaluation

We evaluate the performance of the proposed data compression framework through simulations. We first investigate the effectiveness of the EDM scheme by comparing its predicted results with the actual joint coding performance of practical coding schemes. Then, we study the compression performance of DMCP under changing network sizes and camera settings.

3.5.1 Validity of the EDM Predictions

For a cluster of N camera sensors with observations $X_1, \cdots, X_N$, the joint entropy $H(X_1, \cdots, X_N)$ is a theoretical lower bound of the total coding rate for these cameras. To predict the percentage of rate savings of joint coding, we define an *estimated joint coding efficiency* as

$$\eta_H = 1 - \frac{H(X_1, \cdots, X_N)}{H(X_1) + \cdots + H(X_N)} \tag{34}$$

where $H(X_1) + \cdots + H(X_N)$ corresponds to the total coding rate needed when the cameras compress their observations individually.

We verify the estimated joint coding efficiency using practical video coding experiments. Similar as the definition above, we introduce an *actual joint coding efficiency* as

$$\eta_R = 1 - \frac{R(X_1, \cdots, X_N)}{R(X_1) + \cdots + R(X_N)} \tag{35}$$

where $R(X_1, \cdots, X_N)$ is the total rate from joint coding, and $R(X_1) + \cdots + R(X_N)$ is the total rate from individual coding.

Table 4: Parameters for entropy-based divergence measure

	H.264 Baseline	H.264 MVC
RD optimization	on	on
Entropy coding	UVLC	CABAC
Search range	128 pixels	128 pixels
Num of reference frames	1	1

We consider an indoor scene and an outdoor scene as representatives of various WMSN applications. We deploy 12 camera nodes at different view points around each scene, and let each camera capture one image of the scene, with a resolution of 512 ×384. Fig. 14 shows two of the captured images. We record each camera's FoV parameters, so that the joint entropy and the corresponding η_H can be estimated using EDM. We also perform joint coding on the captured images, and from the resulting coding rates we can obtain η_R.

Experiments on different cluster sizes, coding schemes, and coding parameters are performed to evaluate the joint coding efficiency. The cluster size is set to three different values (N=2, 3, and 4). As for coding schemes, there are many standardized solutions such as the JPEG/JPEG 2000 and the MPEG/H.26x series. For joint coding on multiple images, the redundancy among different images should be removed. The JPEG/JPEG 2000 standards can only reduce the redundancy within a single image, thus they are not suitable for use here. We use two coding schemes of the H.264 standards: the Baseline profile and the recently developed Multi-View Coding (MVC) extension. And the H.264 reference softwares JM 8.5 [2] and JMVC 2.5 [3] are used, respectively. For both coding schemes, we obtain the coding rates under three quantization steps (QP=28, 32, and 37). Other key parameters in the coding experiment are listed in Table 4.

In (35), the rates of individual coding are obtained by performing intra coding on each image in the cluster, while the rate of joint coding are obtained by performing

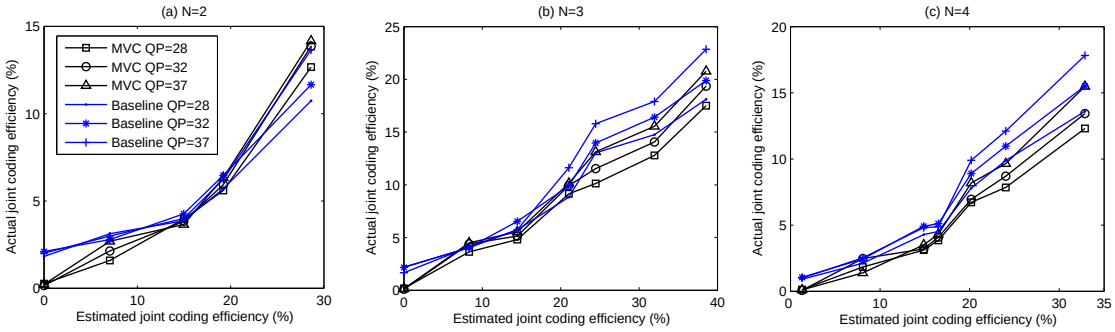

Figure 15: Joint coding performance of the indoor scene "Tables".

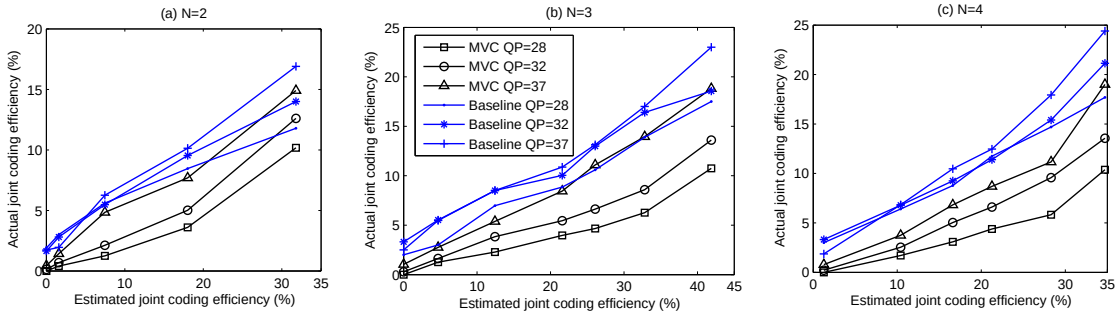

Figure 16: Joint coding performance of the outdoor scene "Trees".

predictive coding among the images. For predictive coding, the images in the cluster are coded in a sequential order. We also use the dependency graphs in the EDM algorithm to guide the coding process. In a dependency graph each camera is connected with the camera that is most correlated with it, thus, it is beneficial to perform predictive coding between cameras that are connected in the graph. For example, if three cameras have a dependency graph as $C_1 \rightarrow C_2 \rightarrow C_3$, for joint coding of the images $\{X_1, X_2, X_3\}$, we take X_1 as the reference image and encode it first, and then encode X_2 based on the prediction of X_1, and X_3 based on the prediction of X_2. The total joint coding rate is a sum of the coding rates of the three images.

When deploying the cameras around a scene, we let their locations and sensing directions be pairwisely symmetric with respect to the center of the scene. Consequently, we can have several (at least two) groups of cameras that lead to the same estimated joint coding efficiency (η_H), according to our spatial correlation model and

the EDM algorithm. For each value of η_H, we perform joint coding on the corresponding groups of cameras, and take the average value of the resulting actual coding efficiencies (η_R). Comparisons of the corresponding η_H and the average η_R values for the two scenes are given in Fig. 15 and Fig. 16. For both scenes, although the actual joint coding efficiency might be smaller than the estimated joint coding efficiency, the actual joint coding efficiency increases as the estimated joint coding efficiency increases.

As shown in Fig. 15 and Fig. 16, for the same coding scheme, the value of η_R increases as the quantization step increases: as larger quantization steps result in more distortion, they may have more potential bit savings for joint coding. In particular, compared to the indoor scene "Tables", the outdoor scene "Trees" contains more details such as the textures in the tree leafs and the grass fields. Therefore, the coding performance of the outdoor scene is more sensitive to the extent of quantization. As shown in the figures, the results for the outdoor scene have more deviation when the quantization step varies. The H.264 MVC extension is more advanced than the H.264 Baseline profile, and our experiments also show that the MVC extension always produces lower bit rates under the same coding parameters. However, the joint coding efficiency of the MVC extension is not necessarily larger than that of the Baseline profile. This is because the MVC extension results in smaller denominators in (35) than the Baseline profile.

In general, the actual joint coding efficiency is proportional to the estimated joint coding efficiency, and such feature is independent of cluster sizes, coding methods, and levels of distortion. Therefore, the EDM scheme can effectively predict the joint coding performances for different sets of cameras for typical applications of wireless multimedia sensor networks.

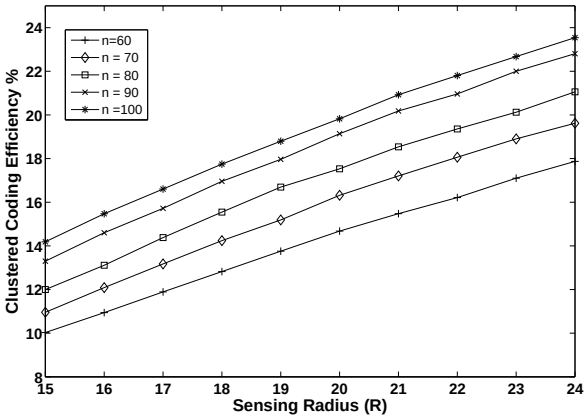

Figure 17: Compression performance vs. network size n and sensing radius R

3.5.2 Compression Performance of DMCP

We now study the compression performance of DMCP in terms of clustered coding efficiency, which has the form similar to equation (34), except that the joint entropy in the entire network is equal to the total entropy produced in the entire network after DMCP is performed. We consider a network with camera sensor nodes uniformly deployed in a 100×100 region. We vary the network size n and sensing radius R, and measure the cluster coding efficiency in Fig. 17. We observe that the DMCP incurs up to 10% - 23 % coding rate reduction in WMSNs. The increase in the clustered coding efficiency under larger sensing radius can be attributed to the following: larger sensing radius leads to higher probability of two adjacent nodes having overlapped FoVs, thus inducing more visual redundancy in the network. The DMCP ensures that these increased redundancy can be effectively identified and removed, thus giving a better compression performance. We also observe that the increase in the number of nodes does not impact the coding efficiency significantly, and thus the DMCP provides good compression scalability.

We now study the impact of sensing direction $\vec{V}$ and offset angle α on the compression performance of DMCP. The deviation in the sensing directions of multiple

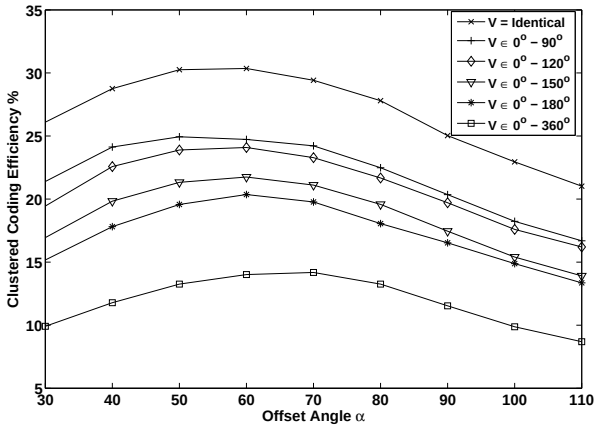

Figure 18: Compression performance vs. sensing direction $\vec{V}$ and offset angle α

camera sensors directly affects the similarity among their retrieved images. For a group of sensors with similar sensing directions, there is high probability that they may capture the similar visual content, thus leading to more redundancy in the network. The DMCP ensures that the sensor nodes with similar directions are grouped together, aiming to reduce the redundancy to the maximum extent. Fig. 18 depicts the coding efficiency of DMCP under changing sensing direction patterns. Here, each sensor node is randomly assigned a sensing direction within a degree region, and wider region leads to larger direction deviation. We observe that a substantial coding efficiency (10% - 15 %) is achieved even in the worst scenario, e.g., each sensor randomly selects a direction within a region of $0° - 360°$, while the optimal coding scenario (20% - 29%) occurs when all the cameras have identical sensing directions.

Besides sensing direction, offset angle also has significant impact on compression efficiency. In Fig. 18, as the offset angle increases, we observe the elevation in coding efficiency, followed by a gradual decrease. This phenomenon is attributed to the following: a wide offset angle leads to a large FoV. Thus, there is greater probability that adjacent cameras cover a large common area. This indicates that more redundancy exists in the network. Therefore, higher compression performance is achievable by DMCP. When the offset angle is over a threshold, e.g., $60° - 70°$ in

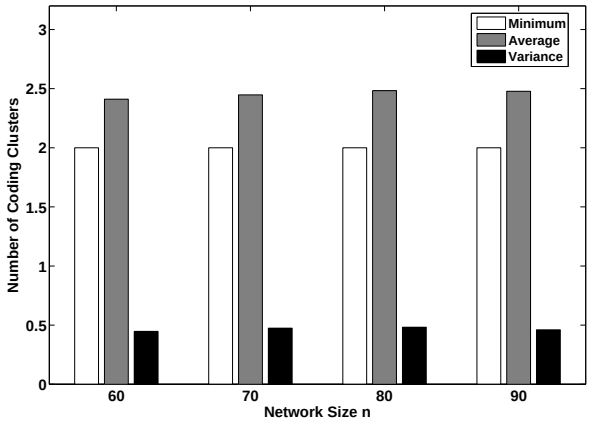

Figure 19: Average and minimum number of cluster heads covering each node

Fig. 18, the increase in offset angle leads to larger size of nonoverlapped FoVs than overlapped ones, thus incurring a reduced compression efficiency.

We now investigate the decoding reliability of DMCP by examining the minimum and average number of CHs covering each camera sensor. As shown in Fig. 19, the minimum number of CHs for each sensor is 2. Meanwhile, we observe that the average number of CHs covering each node exceeds 2. This indicates that some camera sensors are included in more than 2 coding clusters, thus providing additional decoding robustness at data sink. In addition, low variance in the number of CHs is shown in Fig. 19, which proves the fairness of DMCP in terms of coverage performance.

We next compare DMCP to the hybrid energy-efficient distributed clustering (HEED) protocol [74] and its modified version MHEED. HEED is a well known clustering protocol that is specially designed for wireless sensor networks that deal with scalar data. This protocol constructs a hierarchical network architecture by two phases: CH selection and cluster member assignment. In the first phase, sensor nodes are selected as CHs probabilistically. More specifically, each node is given a initial probability p (i.e., 0.05 in [74]) with which it becomes a CH. In the first iteration, each sensor uniformly draws a value between 0 and 1 and compares this value with the initial probability. If this value is less than p, the sensor becomes a CH and all its

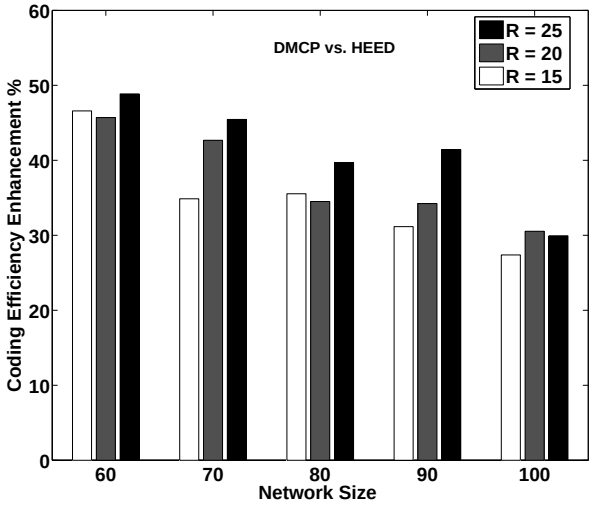

Figure 20: Coding efficiency of DMCP compared with HEED

neighbors are covered. After this iteration, many sensors may still be uncovered since the initial probability (i.e., 0.05) is very small. Therefore, in each of the following iterations, every sensor doubles p and with this probability the uncovered sensors become new CHs. When p reaches 1, the first phase completes. In the second phase, each sensor is assigned to the closest CH as its cluster member. Different from DMCP, HEED protocol is a compression-unaware approach. To fairly compare DMCP with HEED, we design a modified HEED (MHEED) by incorporating the proposed entropy-based divergence measure (EDM) scheme. Specifically, MHEED uses the same procedure as HEED for the CH selection phase. In the second phase, we use the average cluster entropy, instead of node proximity to the CHs, as the metric to associate sensors with CHs. That is, each sensor joins the cluster with the minimum average entropy, a ratio of the estimated joint entropy of the cameras covered by a CH to the number of cameras it covers.

In Fig. 20 and 21, we measure the coding efficiency of HEED and MHEED, respectively and evaluate the coding efficiency enhancement of DMCP, compared with HEED and MHEED, varying the network size n and sensing radius R. Since

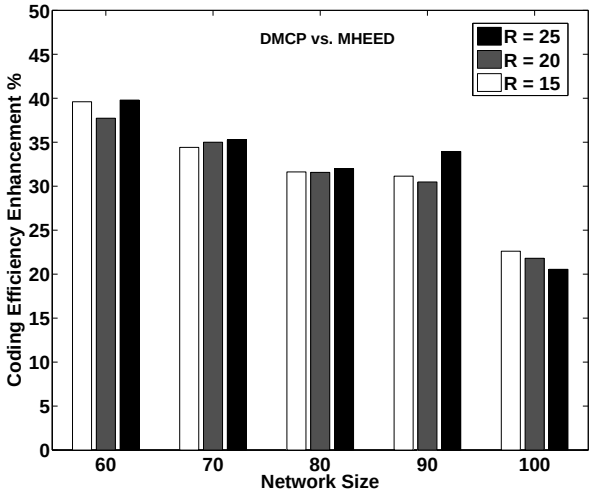

Figure 21: Coding efficiency of DMCP compared with modified HEED

DMCP exploits the inherent correlation structure of multiple cameras, it is expected that DMCP can achieve higher coding efficiency by finely identifying and properly selecting a set of clusters that leads to higher compression performance. Accordingly, as shown in Fig. 20 and 21, DMCP achieves $28\% - 50\%$ and $20\% - 40\%$ enhancement, compared with the coding efficiency of HEED and MHEED, respectively. Meanwhile, in Fig. 20 and 21, we observe that higher enhancement is achieved under smaller network size (i.e., smaller camera density because of the fixed deployment area). This phenomenon is attributed to the fact that smaller camera density leads to higher variability of the joint entropy of different clusters. In this case, the correlation based strategy, DMCP, has more evident advantage over compression-unoriented approaches like HEED and MHEED. Moreover, we observe that less enhancement is achieved in Fig. 21 than in Fig. 20. This implies that MHEED achieves higher coding efficiency than HEED. This is as expected because MHEED uses the average entropy as the metric in selecting CHs, which is superior to just selecting the closest CH, because the average entropy of a node is a measure of the expected coding performance if this node is selected as CH. It is also worth to notice that in Fig. 21 the enhancement under different sensing radius settings is comparable, which indicates MHEED is

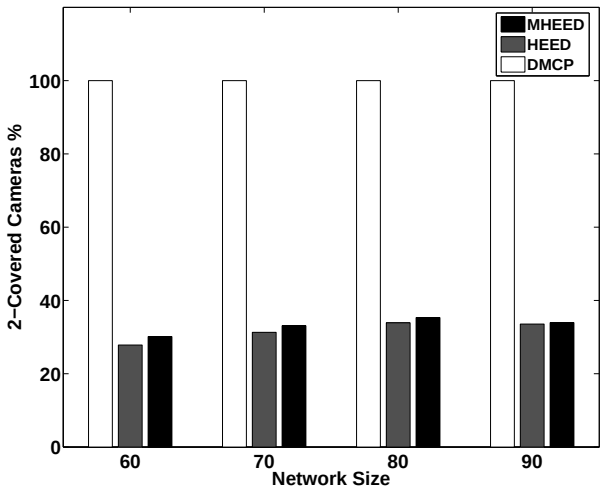

Figure 22: Percentage of cameras covered by more than 2 clusters

less sensitive to the changing sensing radius than HEED because MHEED partially exploits the correlation of visual information in clustering procedures.

We now investigate the reliability of HEED and MHEED by examining the percentage of 2-covered cameras, i.e., the cameras that are covered by more than two clusters. As shown in Fig. 22, the percentage of 2-covered cameras under HEED and MHEED is around 30%, comparing with 100% under DMCP. This implies that DMCP establishes a more robust coding hierarchy than HEED and MHEED. Meanwhile, we also observe a slight elevation in the percentage of 2-covered cameras under HEED and MHEED as the network size or node density increases. This is due to the fact that higher node density gives rise to more 2-covered cameras. This could further increase the percentage of cameras covered by more than two clusters in the network.

CHAPTER IV

CORRELATION-BASED SCHEDULING

4.1 Introduction

MAC layer protocols for WMSNs should enable energy-efficient channel access policies and differentiated scheduling of heterogeneous traffic to support application-specific QoS requirements [5]. Based on the channel access policies, MAC protocols could be classified into contention-based protocols and contention-free protocols. Contention-based protocols are mostly based on variants of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. For example, the S-MAC [73] and the T-MAC [17] protocols are in this type. These protocols alternate between sleep cycles and listen cycles to save energy in sensor networks, but energy saving is accomplished at the cost of latency and by allowing throughput degradation. Some contention-based MAC protocols also provide differentiating network services based on priority levels to satisfy QoS requirements [54]. However, there is little performance guarantee due to the random access nature of contention-based protocols. Contention-free protocols are primarily based on reservation of time slots or channels. The Time Division Multiple Access (TDMA) is a representative protocol of this class, in which the cluster head or sink helps in slot assignment, querying particular sensors and maintaining time schedules. There have been studies on TDMA for sensor networks, such as how to perform slot allocation [48] and how to customize TDMA for different communication patterns in sensor networks [35]. We believe that contention-free protocols will be more promising for providing QoS support in WMSNs.

In this work, we consider a clustered network architecture consisting of powerful multimedia processing hubs as cluster heads and ordinary camera sensors as

cluster members. Within each cluster, camera sensors report their observations to the multimedia hubs following TDMA scheduling, while the communication between multimedia hubs is through different spectrum channels. We believe that such architecture can facilitate QoS provisioning in WMSNs. Using this network architecture, we propose a scheduling framework that finds the optimal schedules of camera sensors by leveraging the correlation of visual information. The design principles of this framework are explained as follows.

The images observed by cameras with overlapped FoVs are correlated, thus leading to substantial redundancy in the network traffic. To remove such redundancy, camera sensors can perform inter-camera differential coding with each other by allowing a camera to encode its image based on the reference of the image from another camera, so that the coding rate could be reduced. This differential coding rate depends on the degree of the correlation between the two cameras. As introduced in Chapter 2, the correlation of visual information among multiple cameras is explicitly measured by the sensing parameters and locations of cameras, which are independent of image and coding algorithms. By leveraging this unique characteristic, we can introduce differential coding in the network scheduling process to reduce the redundancy in the network.

We propose a scheduling framework consisting of three fundamental problems [66]. The first problem is how to construct a scalable network architecture that improves spectrum utilization. In a WMSN, a multi-tier network architecture is recommended [5]. For example, many WMSN testbeds, such as the SensEye [34] and the IrisNet [47], adopt multi-tiered network topology, and it has been shown to improve energy efficiency. Therefore, we propose to partition the energy-constrained camera sensors into multiple clusters with each cluster coordinated by a multimedia processing hub, which is equipped with higher communication and processing capabilities. Under this network architecture, the network throughput is enhanced by applying the concept

of frequency reuse, which allows concurrent transmissions within multiple clusters. However, in a WMSN, the effectiveness of frequency reuse may be jeopardized by the constrained resource of camera sensors. More specifically, the number of available orthogonal channels that camera sensors can switch to is limited by their hardware specifications and the spectrum availability. On the other hand, vertex coloring theorems [27] imply that the number of orthogonal channels should exceed the maximum number of neighboring clusters in a network to guarantee that all neighboring clusters can be assigned with different channels, Therefore, to increase network throughput of a WMSN, placing hubs at proper locations that facilitate frequency reuse is of paramount importance.

After hubs are located, our second problem is how to assign each camera to a hub in such a way that the overall image compression efficiency is enhanced. Specifically, we consider a joint coding-based camera assignment approach (JCA). In JCA, each hub acts as a single encoder and perform joint coding on the images collected from multiple cameras. The coding rate a hub can achieve depends on the correlation among its member cameras. Specifically, associating a hub with a group of cameras having high correlation can remove a substantial amount of redundancy and lead to small coding rate. Therefore, the design of a correlation-based assignment strategy, which optimizes the global compression performance, is another primary task in WMSNs.

After cameras are assigned to proper hubs, our third problem is how to design an image gathering schedule within each cluster so that the network lifetime is increased. Specifically, we design a differential coding-based scheduling approach (DCS). In DCS, a camera is allowed to wake up at a certain time slot and overhear the on-going transmission of a neighboring camera. After that, it encodes its own image conditionally based on the prediction of the previously overheard image, and sends its image with a reduced coding rate. The differential coding rate a camera can generate depends on

the degree of the correlation between this camera and the one whose image it overhears. Thus, the design of a correlation-oriented schedule, which significantly reduces the differential coding rates, helps to prolong the network lifetime.

The rest of this chapter is organized as follows. Section 4.2 mathematically formulates the problems. In Section 4.3, we propose an algorithm to find the optimal locations to place the multimedia processing hubs. The camera assignment problem is investigated in Section 4.4, and the problem of scheduling within a cluster is studied in Section 4.5. The performance of the scheduling framework is evaluated in Section 4.6.

4.2 Problem Formulation

We propose a correlation-based scheduling framework to efficiently gather the images generated by camera sensors. This framework consists of three components including MinMax Degree Hub Location (MDHL), Minimum Sum-entropy Camera Assignment (MSCA), and Maximum Lifetime Scheduling (MLS). The MDHL problem aims to find the optimal locations to place the multimedia processing hubs, which operate on different channels for concurrently collecting images from adjacent cameras, such that the number of channels required for frequency reuse is minimized. With the locations of the hubs determined by the MDHL problem, the objective of the MSCA problem is to assign each camera to a hub in such a way that the global compression gain is maximized by jointly encoding the correlated images gathered by each hub. At last, given a hub and its associated cameras, the MLS problem targets at designing a schedule for the cameras such that the network lifetime of the cameras is maximized by letting highly correlated cameras perform differential coding on the fly.

4.2.1 Correlation-based Joint Coding and Differential Coding

To remove the redundancy among correlated camera sensors, a group of camera sensors can collaboratively compress their data by joint coding and differential coding.

Consider a cluster consisting of a multimedia hub with high processing capabilities and N ordinary camera sensors $\{v_1, \ldots, v_N\}$, where each camera v_i produces image X_i. We can perform *multi-camera joint coding* in the cluster: each camera sends its individual images to the hub, while the hub acts as a single encoder that takes all collected images as inputs and perform joint coding. We denote the total coding rate of all the images by $R(X_1, \cdots, X_N)$. According to Shannon's source coding theorem, the total coding rate of all nodes within a cluster is lower bounded by the joint entropy of the observations $H(X_1, X_2, \ldots, X_N)$, given by $R(X_1, \cdots, X_N) \geq H(X_1, X_2, \cdots, X_N)$.

On the other hand, two camera sensors can also perform *inter-camera differential coding* with each other. For two images X_i and X_j observed by cameras v_i and v_j, we can compress X_i based on the prediction of X_j. We denote the resulting differential coding rate of X_i by $R(X_i|X_j)$, and $R(X_i|X_j)$ satisfies $R(X_i|X_j) \geq H(X_i|X_j)$, where $H(X_i|X_j)$ is the conditional entropy of X_i given the knowledge of X_j. The conditional entropy can be derived from joint entropy as $H(X_i|X_j) = H(X_i, X_j) - H(X_j)$.

Our previous results in Chapter 2 and Chapter 3 show that the joint entropy for multiple images can be effectively estimated based on the correlation between cameras, and this correlation is given by a function of camera settings before the actual images are captured. Specifically, if two cameras C_j and C_k can both observe an area of interest P_i, a spatial correlation coefficient $\rho_{j,k}$ for the observations of P_i at C_j and C_k is derived as

$$\rho_{j,k} = f(O_j, \vec{V_j}, O_k, \vec{V_k}, P_i) \tag{36}$$

which indicates that $\rho_{j,k}$ is a function of the two cameras' locations (O_j, O_k) and sensing directions ($\vec{V_j}$, $\vec{V_k}$) as well as the location of the area of interest P_i.

4.2.2 MinMax Degree Hub Location Problem

Consider a camera network modeled by a graph $G = (V, E)$, where V is a set of cameras, i.e., $V = \{v_1, v_2, ...v_n\}$, and E is a set of links. A link (v_i, v_j) exists if v_i and

v_j are within 1-hop range of each other.

Definition 7. The degree of a hub h*, denoted by $deg(h)$, is the total number of hubs (except h) that reside within the 2-hop range of the hub h.*

To facilitate frequency reuse, the neighboring clusters must be assigned with different channels and the cameras must be able to operate on the channels of their associated clusters. Since the maximum distance between two neighboring clusters is 2-hop distance, by graph coloring theorems [27], this implies that the maximum degree of hubs should be less than the available orthogonal channels to ensure the effectiveness of frequency reuse. For this purpose, we define the MinMax Degree Hub Location Problem as follows

Definition 8. MinMax Degree Hub Location Problem (MDHL)*: given a graph $G = (V, E)$ and a set of potential hub locations $F = V$, find a subset $F' \subseteq F$ such that the maximum degree of hubs, $\max_{h \in F'}(deg(h))$, is minimum, and for all $v_i \in V$, there is at least one hub $h \in F'$ for which $(h, v_i) \in E$.*

4.2.3 Minimum Sum-entropy Camera Assignment Problem

After the optimal locations of hubs are determined, our next task is to associate cameras with proper hubs. Since joint coding rate is lower bounded by joint entropy, we formulate the camera assignment task by introducing an optimization problem, namely, Minimum Sum-entropy Camera Assignment Problem (MSCA). Towards this, we model the camera assignment choices for a hub as a collection of subsets of cameras within the hub's 1-hop neighborhood, with each subset associated with a weight, which is the joint entropy of the cameras in this subset. We adopt the following notations.

- S_h^e: set of cameras exclusively covered by h.
- S_h^c: set of cameras commonly covered by h and other hubs.

- $P(S_h^c)$: power set of S_h^c, which is a set of all subsets of S_h^c, including empty set and S_h^c itself.

- A_h: assignment choice set of h, $A_h = \{S_h^e \bigcup W\}_{W \in P(S_h^c)}$.

Apparently, each hub h may have exponentially many choices (i.e., $|A_h| = |P(S_h^c)| \leq 2^{|V|}$), and we aim to choose one choice (subset) for each hub so that the total costs of the choices (the total entropy of the subsets) is minimum. Now, the Minimum Sum-entropy Camera Assignment problem (MSCA) is formally defined as follows.

Definition 9. Minimum Sum-entropy Camera Assignment problem (MSCA)*: given a set V of cameras, k hubs, a collection of assignment choice sets of centers, $S = \{A_{h_i}\}_{i=1}^{k}$, with each assignment choice set associated with a nonnegative weight , find a minimum weight $S' \subseteq S$ of cardinality k, which covers all elements in V.*

4.2.4 Maximum Lifetime Scheduling Problem

Given a hub and its member cameras, each hub will generate an order to schedule images collections from its members. Our task is to find the optimal schedule such that the lifetime of the member cameras is maximized.

Definition 10. The lifetime of the member cameras *is the time duration when all the members of a hub keep alive.*

Assume that cameras have equal initial energy. The maximization of the lifetime of the cameras in a cluster is equivalent to minimization of the maximum energy consumption of the cameras in this cluster. Let $E_{tx}(h, v_i)$ denote the energy consumed by v_i to convey its image to h. $E_{tx}(h, v_i)$ is a function of $\{d_{(h,v_i)}, R_{v_i}\}$, in which $d_{(h,v_i)}$ is the Euclidean distance between h and v_i and R_{v_i} is the predicted differential coding rate of v_i. Consequently, we formulate the Maximum Lifetime Scheduling Problem (MLS) as follows.

Definition 11. Maximum Lifetime Scheduling Problem (MLS)*: given a hub h and a set A_h of cameras assigned to h, find a schedule σ assigning a pair of slots for each cameras to transmit and overhear in such a way that the maximum energy consumption,* $\max_{v_i \in A_h} E_{tx}(h, v_i)$*, is minimum.*

In the following sections, we analyze the complexity of these problems, and then propose approximation and heuristic algorithms to solve them.

4.3 MinMax Degree Hub Location Problem

In this section, we first prove that MDHL is NP-complete. Next, we formulate MDHL problem as an integer program (IP). Then, we present an approximation algorithm by applying the linear relaxation and random rounding technique, which was originally studied in MAX-2SAT [24] and Covering & Packing problems [52].

4.3.1 NP-completeness

First, the decision version of the MDHL is as follows.

Definition 12. Decision Version of MDHL*: given a graph $G = (V, E)$, a set of potential hub locations $F = V$, and a positive integer k, determine if there exists a subset $F' \subseteq F$ with the maximum degree of hubs,* $\max_{h \in F'}(deg(h)) \leq k$ *such that for all $v \in V$, there is at least one hub $h \in F'$ for which $(h, v) \in E$.*

Theorem 7. *The MDHL is NP-complete.*

Proof. First, we argue that the decision version of MDHL $\in$ NP since given a instance of MDHL, a verification algorithm can efficiently check if each camera has at least one hub in its neighborhood, and if the maximum degree of hubs is k. Thus, the MDHL belongs to NP.

We will now show that the Minimum Dominating Set problem (MDS) is polynomial time reducible to MDHL, i.e., MDS $\leq_P$ MDHL. An instance of MDS is given

by a graph $\overline{G} = (\overline{V}, \overline{E})$, and a positive integer $k-1$. The objective is to determine if there exists a dominating set $\overline{V'} \subseteq \overline{V}$ such that $\left|\overline{V'}\right| \leq k-1$ and each element $v \in \overline{V}$ is a neighbor of at least one element of $\overline{V'}$.

Next, we will construct an instance of MDHL problem from an instance of MDS. We define sets V, F, E as follows: let $V = \overline{V} \bigcup \{u, f'\}$, where u and f' are new elements; Let $F = \overline{V} \bigcup \{f'\}$; Establish a link between u and each element in $\overline{V}$ as well as a link between u and f', and add these links to $\overline{E}$, i.e., $E = \overline{E} \bigcup \{(u, f')\} \bigcup \{(u, v)\}_{v \in \overline{V}}$. Then, the instance of MDHL is given by a graph $G = (V, E)$, a set F, and a positive integer k.

We will now prove that the original instance of MDS is a yes instance if and only if the MDHL instance we created is also a yes instance. First, suppose the instance of MDHL has a solution $F' \subseteq F$ with $\max_{h \in F'}(deg(h)) \leq k$. By our construction, f' is the only 1-hop neighbor of u, and u is the 1-hop neighbor of every element in V. It indicates that f' is the 2-hop neighbor of every element in V. Moreover, since $u \notin F'$, f' will be added in F' to cover itself. This implies that f' is the element in F' that has the maximum degree k. Meanwhile, since $\overline{V} = V - \{f'\}$, this indicates that the instance of MDS has a dominating set $\overline{V'} \subseteq \overline{V}$ of cardinality less than $k-1$. Next, suppose that there is a dominating set $\overline{V'} \subseteq \overline{V}$ with $\left|\overline{V'}\right| \leq k-1$ in the original MDS instance. By the similar arguments, the degree of the elements in F' is at most k in the constructed MDHL instance.

We now have shown that MDS problem can be solved by the proposed construction and an algorithm that solves MDHL. Since our construction takes polynomial time, and MDHL is NP, we can conclude that MDHL is NP-complete. □

4.3.2 IP Formulation of MDHL

We first model the MDHL as an integer nonlinear program (INP). Consider a camera network described by a graph $G = (V, E)$ and a set of potential hub locations $F = V$.

First, we define 1-hop neighborhood and 2-hop neighborhood of a camera $v_i \in V$, respectively.

Definition 13. The 1-hop neighborhood of v_i, *denoted by* S_i^1, *is a set consisting of* v_i *and cameras within 1-hop range of* v_i.

Definition 14. The 2-hop neighborhood of v_i, *denoted by* S_i^2, *is a set of cameras within 2-hop range of* v_i.

We assign a variable x_i for each camera $v \in V$, which is allowed 0/1 values. This variable will be set to 1 iff a hub is placed at the location of v_i. Consequently, the MDHL problem can be formulated as a Integer Nonlinear Program INP_{MDHL}

$$\text{MIN} \quad y \tag{37}$$

$$s.t \quad \sum_{j:v_i \in S_j^1} x_j \geq 1, \quad \forall v_i \in V \tag{38}$$

$$\sum_{j:v_i \in S_j^2} x_i x_j \leq y, \quad \forall v_i \in V \tag{39}$$

$$x_j \in \{0,1\}, \qquad \forall v_j \in V \tag{40}$$

The objective function y is the maximum degree of all hubs ($\{v_i | x_i = 1\}$). The first constraint states that each camera $v_i \in V$ must reside within the 1-hop neighborhood of at least one hub, whereas the second constraint indicates that the degree of each hub (described in Definition 1) must be less than the maximum value. As the second constraint (39) is quadratic, the formulated integer program INP_{MDHL} is not linear. To linearize INP_{MDHL}, the quadratic constraint (39) is eliminated by applying the techniques proposed in [22]. More specifically, the product $x_i x_j$ is replaced by a new binary variable w_{ij}, on which several additional constraints are imposed. As a consequence, we can reformulate INP_{MDHL} exactly to a integer linear Program

IP_{MDHL} by introducing the following linearlization constraints:

$$\sum_{j:v_i \in S_j^2} w_{ij} \leq y, \ \forall v_i \in V \tag{41}$$

$$w_{ij} \leq x_i, w_{ij} \leq x_j, \ \forall v_i, v_j \in V \tag{42}$$

$$w_{ij} \geq x_i + x_j - 1, \ \forall v_i, v_j \in V \tag{43}$$

$$w_{ij} \geq 0, \ \forall v_i, v_j \in V \tag{44}$$

and removing the quadratic constraint (39). By relaxing variables $x_i \in \{0, 1\}$ to $x_i \geq 0$, we get the relaxed linear program LP_{MDHL} consisting of the objective function (37) along with constraints (38), (41), (42), (43), (44), and $x_i \geq 0, \ \forall v_j \in V$.

4.3.3 Randomized Approximation Algorithm

Given an instance of MDHL modeled by the integer program IP_{MDHL}, the proposed algorithm (see *Algorithm* 7) is the following: first solves the relaxed linear program LP_{MDHL} to get an optimal fractional solution, denoted by $(\mathbf{x}', y')$, where $\mathbf{x}' =< \mathbf{x}'_1, \mathbf{x}'_2, ...\mathbf{x}_{|V|'} >$, and round x'_i to integers $\overline{x}_i$ by a random rounding procedure. This procedure consists of three steps: (i) first set all $\overline{x}_i$ to be 0; (ii) then let $\overline{x}_i = 1$ with probability x'_i and execute this step for log(n)+2 times.

Algorithm 7 Approximation Algorithm for MDHL

1: Solve LP_{MDHL}. Let $(\mathbf{x}', y')$ be the optimum solution.
2: $\overline{\mathbf{x}} \leftarrow \mathbf{0}$, $j \leftarrow 0$.
3: **while** $t \leq log(n) + 2$ **do**
4: $\quad \overline{x}_i \leftarrow 1$ with probability $\overline{p}_i \leftarrow x'_i$, $t \leftarrow t + 1$
5: **end while**
6: Return $(\overline{\mathbf{x}}, \overline{y})$

Theorem 8. *Let OPT denote the optimal solution of the MDHL problem. The proposed algorithm yields a solution of* $O(log^2(n))OPT$ *in expectation*

Proof. Let y^* denote optimal solution to the MDHL problem. Consider any element

$v_i \in V$. Its expected degree follows

$$\mathrm{E}(\sum_{j:v_i \in S_j^2} \overline{x}_i \overline{x}_j) = \mathrm{E}(\overline{x}_i)\mathrm{E}(\sum_{j:v_i \in S_j^2} \overline{x}_j) = \mathrm{E}(\overline{x}_i) \sum_{j:v_i \in S_j^2} \mathrm{E}(\overline{x}_j) \tag{45}$$

The first equality holds because v_i is not in its own 2-hop neighborhood S_i^2 by Definition 9 and thus $\overline{x}_i$ and $\overline{x}_j$ are independent. The second equality holds because of linearity of expectation. Applying union bound, we have the probability that an element becomes a hub (i.e., $\overline{x}_i = 1$) when the random rounding is done

$$\Pr[\overline{x}_i = 1] = \Pr[\bigcup\nolimits_{t \leq \alpha+1} \overline{x}_i = 1 \text{ at round t}] \leq \alpha x_i'$$

where $\alpha = log(n) + 2$. This implies $E(\overline{x_i}) \leq \alpha x_i'$. Letting $\delta = (1 + 1/\Delta)^2$, we obtain the expected degree of v_i

$$\mathrm{E}(\sum_{j:v_i \in S_j^2} \overline{x}_i \overline{x}_j) \leq \alpha^2 \sum_{j:v_i \in S_j^2} x_i' x_j' \leq \alpha^2 y',$$

which implies $\mathrm{E}(\sum_{j:v_i \in S_j^2} \overline{x}_j) \leq \Delta\alpha^2(y' + \delta)$.

Next we consider the probability that an element $v_i \in V$ has no hub in its 1-hop neighborhood at round j, that is,

$$\prod_{j:v_i \in S_j^1} Pr[\overline{x_j} = 0 \text{ at round j}] = \prod_{j:v_i \in S_j^1} (1 - x_j')$$
$$\leq \prod_{j:v_i \in S_j^1} e^{-x_j'} = e^{-\sum_{j:v_i \in S_j^1} x_j'} \leq \frac{1}{e}.$$

The first inequality results from the inequality $(1 - x) \leq e^{-x}, \forall x \in [0, 1]$. The second inequality follows the fact that $\sum_{j:v_i \in S_j^1} x_j' \geq 1$. Now, the probability that an element v_i has no hub in its 1-hop neighborhood after the random rounding is $e^{-(log(n)+2)} < 1/4n$ Then, by union bound, we get

$$\Pr[\text{Some element has no neighboring hub}] \leq 1/4. \tag{46}$$

This implies that with probability at least 3/4 the Algorithm 7 yields a solution which is $(\log(n) + 2)^2$ times the solution of the linear program LP_{MDHL}. This completes the proof. □

4.4 Minimum Sum-entropy Camera Assignment

With the knowledge of the hub locations, the MSCA problem aims to associate each hub with a group of cameras of high correlation such that the total joint coding rate of the hubs is minimized. We propose a polynomial time heuristic algorithm to solve the MSCA problem.

4.4.1 Binary Nonlinear Problem Formulation for MSCA

Given a set of hubs $\mathcal{F} = \{h_i\}_{i=1}^{k}$, let a_{ij} be an indicator variable denoting whether camera v_j is assigned to h_i. This value is set to 1 iff v_j is assigned to h_i. Let S_i denote the set of cameras residing within the one-hop range of hub h_i. Consequently, MSCA problem can be formulated as follows.

$$\text{MIN} \quad \sum_{h_i \in F} H(\bigcup_{v_j \in V} a_{ij} X_{v_j}) \tag{47}$$

$$s.t \quad \sum_{j:v_i \in S_j} a_{ij} = 1, \quad \forall v_j \in V \ \forall h_i \in F \tag{48}$$

$$a_{ij} \in \{0, 1\}, \qquad \forall v_j \in V \ \forall h_i \in F \tag{49}$$

The objective function is the total entropy of the whole network. The first constraint states that each camera $v_i \in V$ has to be assigned to exactly one hub.

4.4.2 Polynomial Time Heuristic Algorithm

To solve the MSCA problem, we propose a heuristic algorithm based on greedy approach (see *Algorithm* 8). Given a set of cameras V and a set of hubs, $\mathcal{F} = \{h_i\}_{i=1}^{k}$. At each iteration, we calculate the average entropy of the one-hop neighbor set S_i of each hub h_i and find the set with the minimum average cost among all hubs. If the set S_i is selected, i.e., the cameras in S_i is assigned to h_i, we remove h_i from the collection $\mathcal{F}$ and update the one-hop neighbor set for the remaining hubs by removing the cameras covered by h_i. By analyzing *Algorithm* 8, we have the following lemma.

Lemma 1. *Given k hubs, the proposed algorithm takes k iterations to select k disjoint sets (assignment choices) such that each camera is assigned to exactly one hub.*

In each iteration, we need a subroutine that finds a set S minimizing the ratio of $H(S)/|S|$. Since the joint entropy $H(S)$ of a set of cameras can be estimated in $O(n^2)$ iterations by applying the Entropy-based Divergence Measure (EDM) scheme in Chapter 3, it is easy to prove that the proposed algorithm runs in polynomial time.

Algorithm 8 Heuristic Algorithm for MSCA

1: $C \leftarrow \emptyset$, $\mathcal{F} = \{h_i\}_{i=1}^{k}$
2: **while** $\mathcal{F} \neq \emptyset$ **do**
3: $\quad \{h^*, S_h^*\} \leftarrow \arg\min_{h \in \mathcal{F}} \{H(S_h)/|S_h|\}$.
4: $\quad C \leftarrow C \cup S_h^*$, $\mathcal{F} \leftarrow \mathcal{F} \setminus h^*$, $V = V \setminus S_h^*$, updates $S_i, \forall h_i \in \mathcal{F}$
5: **end while**

4.5 Maximum Lifetime Scheduling Problem

In this section, we first prove that the MLS problem is NP-hard by formulating it as an equivalent binary program. Consequently, we present a randomized approximation algorithm, which produces a solution $\leq OPT + c_{max}/e$ in expectation, where c_{max} is the maximum energy consumed by a camera to send its image to the hub without performing differential coding.

4.5.1 IP Formulation for MLS

Given a hub h and a set A of cameras assigned to it. To save energy, we let the transmission range of each camera $v_i \in A$ be the distance between v_i and hub h, denoted by d_{ih}. For each camera $v_i \in A$, let N_i denote a set of cameras within v_i's transmission range, and let X_i denote the image gathered by v_i. We assign two variables x_i and y_{ji} for each camera $v_i \in A$, which are allowed 0/1 values. x_i is set to 1 iff v_i sends its image without overhearing and performing differential coding. y_{ji} is set to 1 iff v_i overhears v_j's transmission and encodes its image X_i conditional on v_j's image X_j. In particular, y_{ii} is set to 1 iff v_i does not overhear anyone's transmission.

Consequently, we formulate the maximum lifetime scheduling problem as an integer program IP_{MLS}.

$$\text{MIN} \quad z \tag{50}$$

$$s.t \quad \sum_{j:v_i \in N_j} x_j \geq 1, \quad \forall v_i \in A \tag{51}$$

$$\sum_{j:v_i \in N_j} y_{ji} = 1, \quad \forall v_i \in A \tag{52}$$

$$y_{jj} = x_j \geq y_{ji} \quad \forall v_i, v_j \in A \tag{53}$$

$$\sum_{j:v_i \in N_j} y_{ji} H(X_i|X_j) d_{ih}^2 \leq z, \quad \forall v_i \in A \tag{54}$$

$$x_j, y_{ji} \in \{0, 1\}, \quad \forall v_j, v_i \in V \tag{55}$$

The objective function z is the maximum energy consumption of all cameras in A. The constraint (51) ensures that each camera has at least one camera to overhear. The constraint (52) states that each camera only overhears once. The equality of the constraint (53) indicates if v_i decides to send its image without performing differential coding, it will not overhear at all, whereas the inequality $x_j \geq y_{ji}$ states that v_j must send its image before v_i can overhear v_j's transmission. The constraint (54) ensures that the energy consumed by each camera v_i to send its compressed image of $H(X_i|X_j)$ bits over the distance d_{ih} is less the maximum value z. Slightly different from the notation of the classic information theory, we let $H(X_i|X_i) = H(X_i)$, which means that a camera only sends its original image if it does not overhear anyone's transmission, i.e., $y_{ii} = 1$. By relaxing the binary variables $x_j, y_{ji} \in \{0, 1\}$ to $x_j, y_{ji} \in [0, 1]$, we get the relaxed linear problem LP_{MLS}.

4.5.2 Approximation Algorithm for MLS

In this subsection, we propose an approximation algorithm based on the random rounding techniques. More specifically, we call a camera v_i as a broadcaster if its variable $x_i = 1$, and as a listener if $x_i = 0$. The proposed algorithm works as follows

Algorithm 9 Approximation Algorithm for MLS

1: Solve LP_{MLS}. Let $(\mathbf{x}', \mathbf{y}', z')$ be the optimum solution.
2: $\overline{\mathbf{x}} \leftarrow \mathbf{0}$, $j \leftarrow 0$.
3: $\overline{x}_i \leftarrow 1$ with probability x'_i, $\overline{y}_{ii} \leftarrow 1$ if $\overline{x_i} = 1$
4: For each $v_i \in A$ with $\overline{x}_i = 0$, find $O_j = \{v_i | y'_{ji} \neq 0, \overline{x}_j = 1\}$
5: **if** $S_i \neq \emptyset$ **then**
6: $\quad v_{j^*} = \arg\min_{v_j \in O_i} H(X_i|X_j) d_{ih}^2$ and $\overline{y}_{j^*i} \leftarrow 1$
7: **else**
8: $\quad \overline{x}_i \leftarrow 1$ and $\overline{y}_{ii} \leftarrow 1$
9: **end if**
10: Return $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{z})$

(see *Algorithm* 9): initially, let all cameras $v_i \in A$ stay as listeners, after solving the linear problem LP_{MLS}, which yields an optimal fractional solution $(\mathbf{x}', \mathbf{y}', z')$, let each camera $v_i \in A$ become a broadcaster with probability x'_i. Otherwise, the cameras stay as listeners. For each listener v_i, find all broadcasters $v_j \in A$ that have nonzero y'_{ji}, and if such broadcasters exist, assign the listener v_i to the broadcaster having the smallest cost $H(X_i|S_j)d_{ih}^2$, otherwise let v_i become a broadcaster itself.

Theorem 9. *Let OPT denote the optimal solution of the MLS problem. The solution of the proposed algorithm is at most $OPT + H(X)d_{max}^2/e$ in expectation, where d_{max} is the maximum distance between a camera and its assigned hub.*

Proof. First, by the pseudo code in *Algorithm* 8, we can verify that the proposed algorithm produces a feasible solution, that is, when the algorithm is done, every camera is either a broadcaster or a listener. To get the expected energy of a camera v_i, we establishes an overhearing list L_i for v_i, which consists of i's potential broadcasters $(v_j | y'_{ji} \neq 0)$. These potential broadcasters are arranged in an increasing order of the cost $c_{ji} = H(X_i|X_j)d_{ih}^2, j = 1, 2, ..., n$, where n is the list length. By this way, we have

$$y'_{1i} = x'_1, y'_{2i} = x'_2, ..., y'_{ni} = x'_n. \tag{56}$$

The above equalities hold because to reduce the cost of a listener, it has to listen to the broadcaster that leads to the smallest cost as possible as it can. Now, we get the

probability that a camera v_i has no broadcaster in its overhearing list.

$$\begin{aligned} p_i &= \prod_{j:v_j \in L_i} (1 - x'_j) \leq \prod_{j:v_j \in L_i} e^{-x'_j} = e^{-\sum_{j:v_j \in L_i} x'_j} \\ &= e^{-\sum_{j:v_i \in N_j} y'_{ji}} = \frac{1}{e} \end{aligned}$$

The first inequality results from the inequality $(1 - x) \leq e^{-x}, \forall x \in [0, 1]$. The first equality in the second line holds because of (56). The last equality follows the fact that $\sum_{j:v_i \in N_j} y'_{ji} = 1$.

According to the algorithm, if v_i is a broadcaster, an event that occurs with probability x'_i, then v_i has a cost c_{ii}. Otherwise, v_i overhears the first camera in the list. If this camera is a broadcaster, an event that occurs with probability $y'_{ii}y'_{1i}$, then v_i has a cost c_{1i}. If the first camera is not a broadcaster and the second is, an event that occurs with probability $y'_{ii}(1 - y_{1i}{}')y'_{2i}$, v_i has a cost c_{2i}, and so on. If there exists no broadcasters in the list, an event that occurs with probability p_i, then v_i becomes a broadcaster and has a cost less than $c_{max} = H(x)d^2_{max}$. By the fact $\sum_{j:v_i \in N_j} y'_{ji}c_{ji} \leq z' \leq OPT$, the expected cost of a camera has an upper bound

$$\begin{aligned} & y'_{ii}c_{ii} + y'_{ii}y'_{1i}c_{1i} + y'_{ii}y'_{2i}c_{2i} + ... + y'_{ii}y'_{ni}c_{ni} + p_i c_{max} \\ & \leq \sum_{j:v_i \in N_j} y'_{ji}c_{ji} + c_{max}/e \leq OPT + H(x)d^2_{max}/e. \end{aligned}$$

□

Note that the solution of the MLS problem only defines the precedence constraints in the schedule. For example, if $y_{ji} = 1$, this only implies that v_j's transmitting slot must be ahead of v_i's, without specifying v_j's or v_i's slot location in the schedule. Thus, as long as the precedence constraints are satisfied, the cameras' transmitting slots can be arranged in any order.

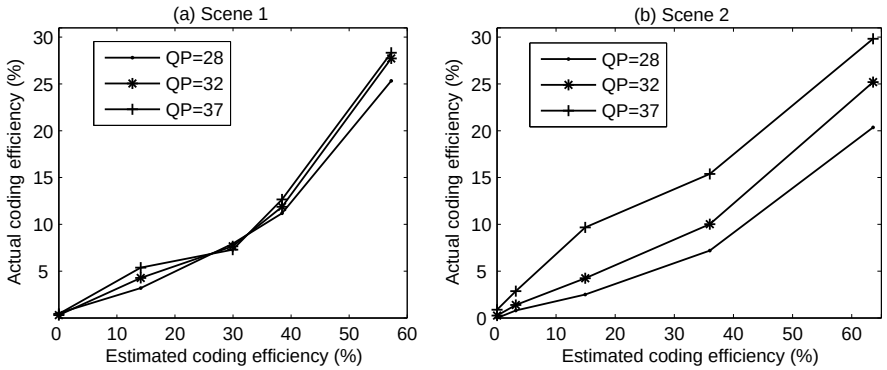

Figure 23: Differential coding efficiency.

4.6 Performance Evaluation

In this section, we evaluate the performance of the proposed schemes. First, we evaluate the effectiveness of the estimator that predicts the efficiency of differential coding between correlated cameras. Then, we study the efficiency of the proposed network deployment approach that consists of the proposed hub placement and camera assignment algorithms. Finally, we evaluate the differential coding-based scheduling algorithm in terms of energy saving.

4.6.1 Validation of the Coding Efficiency Prediction

Since the entropy-based estimator provides predicted coding efficiency for the proposed correlation-based schemes, we need to validate its effectiveness by comparing the estimated coding efficiency with the actual coding efficiency from practical coding experiments. Since the performance of the estimator for predicting joint coding efficiency was tested in the last chapter, we only need to validate its capability to predict the differential coding efficiency. Suppose image X_i is coded based on the prediction of image X_j, and we can define an *estimated differential coding efficiency* as

$$\eta_H^D = 1 - \frac{H(X_i|X_j)}{H(X_i)} \tag{57}$$

Table 5: Parameters for differential coding efficiency prediction

H.264 MVC			
RD optimization	on	Entropy coding	CABAC
Search range	128	Reference frames	1

where $H(X_i|X_j)$ is the theoretical coding rate of differential coding. This metric predicts the percentage of rate savings of differential coding compared to individual coding. The actual differential coding efficiency is calculated by replacing the entropy terms in (57) with the corresponding coding rates from our coding experiment.

In our experiment, we deploy a number of camera nodes in a field and record each camera's FoV parameters. We deploy the cameras in two scenes, an indoor scene and an outdoor one. For each scene, we let each camera capture an image at the same time, and perform coding experiments on the observed images. For any 2 images in the same scene, we take one image as the reference frame and perform multi-view coding on the other image. The H.264 Multi-View Coding (MVC) coding standard with reference software version JMVC 2.5 [3] is used here. To test the performance of differential coding under different parameters, we set three different quantization steps (QP=28, 32, and 37). Other key parameters for the encoder are listed in Table 5. The resulting estimated coding efficiency and actual coding efficiency for the two scenes are plotted in Fig. 23. When the quantization step increases, the actual coding efficiency is slightly higher. This is because a larger quantization step allows for more distortion, in which case more bits could be potentially saved from differential coding. Comparing the results of the two scenes, there is more deviation in coding efficiency for the outdoor scene when the quantization step varies. We find that this is because the outdoor scene contains more texture information, so that the coding performance of the outdoor scene is more sensitive to the extent of quantization. In both cases, the actual differential coding efficiency is approximately proportional

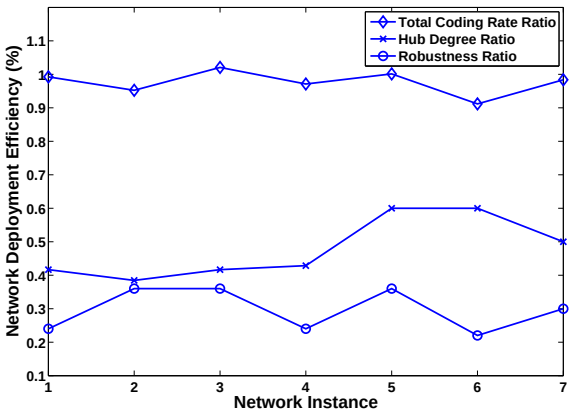

Figure 24: Network deployment efficiency.

to the estimated differential coding efficiency. Therefore, the proposed entropy-based estimation method can be used to predict the performance of inter-camera differential coding.

4.6.2 Efficiency of the Combined Network Deployment Scheme

In this section, we evaluate the joint performance of the proposed hub placement and camera assignment algorithms in terms of spectrum utilization enhancement and image compression efficiency. Specifically, we compare this combined scheme with the DMCP protocol in our previous work in Chapter 3, in terms of the maximum hub degree and the overall coding rate. The DMCP selects a set of clusters that minimizes the total coding rate of all clusters under the assumption that the number of non-overlapped channels is sufficiently large to support parallel transmissions. Here, we consider a WMSN network of 50 camera sensors uniformly distributed in a $100m \times 100m$ region. In Fig. 24, we measure the maximum hub degree and the total coding rate produced by each scheme under 7 randomly generated network topologies, and calculate the ratios of the measurements of the combined scheme to those produced by the DMCP. We observe that the hub degree ratio is less than 0.6, which indicates the combined solution requires much less orthogonal channels than the DMCP, and

thus has capabilities to tolerate the significant reduction of the available channels because of interferences caused by other wireless networks. Moreover, we observe that the total coding rate ratio is almost 1, which implies that both schemes achieve comparable compression efficiency. It is also seen in Fig. 24 that the robustness ratio is less than 1, which means that compared with the DMCP, the combined scheme results in fewer cameras that are covered by more than one hub.

4.6.3 Energy Saving of Differential Coding-based Scheduling

We now investigate the performance of the proposed differential coding-based scheduling scheme. We test the energy efficiency of a cluster by varying the cluster size, deployment range, as well as the FoV parameters of camera sensors.

We consider a cluster with camera nodes uniformly deployed in a 10 $\times$10 meters region. A hub is placed in the center of the region, and each camera node can communicate directly with the hub. To test the performance under different cluster sizes, we deploy 4 to 20 camera nodes within the region. The sensing directions of the cameras are uniformly chosen between 0°- 360°, while the FoV parameters of all the cameras are fixed, with the sensing radius $R = 30$ meters and the offset angle $\alpha = 60°$. For each number of camera nodes, we randomly generate 50 instances and measure the maximum energy consumption per image yielded by our proposed approximated algorithm. As benchmarks, the optimal schedules are also found by the Branch and Bound algorithm, an enumeration based technique. These two algorithms are compared to a conventional TDMA-based scheduling scheme without differential coding between correlated cameras.

The average maximum energy consumption per image for the above schemes are shown in Fig. 25. We observe that the maximum energy of the approximated algorithm is comparable with the optimal solution regardless of cluster sizes. Based on the data in Fig. 25, the average maximum energy of the approximated algorithm is

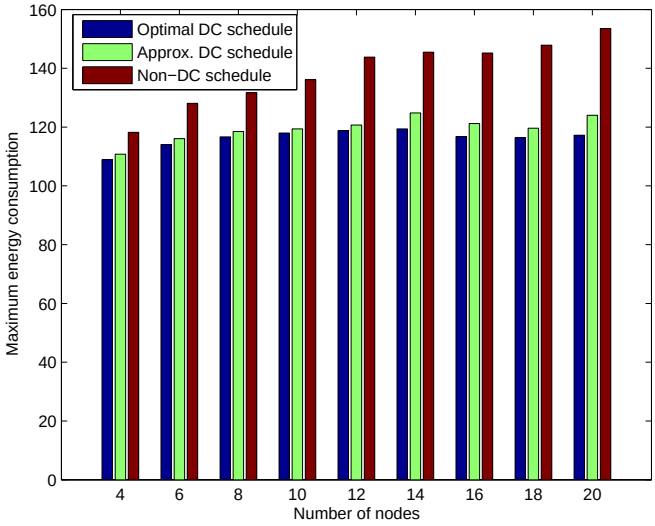

Figure 25: Energy consumption for different cluster sizes.

merely 2.75% more than that of the optimal solution. The approximated scheduling algorithm also leads to 13.68% reduction in terms of average maximum energy consumption compared with the conventional TDMA-based scheduling. This is due to the fact that the differential coding-based scheduling allows cameras to remove the redundancy between each other, thus reducing the bits sent to the hub. Moreover, for the conventional TDMA-based scheduling scheme, the average maximum energy consumption increases as the cluster size increases. In the case that no correlation is exploited, the maximum energy consumption is brought by the node that is farthest away from the hub. Therefore, when the cluster size is large, there is higher probability for a node to be placed far away from the node, so that the average maximum energy consumption is higher. However, as the proposed scheme introduces correlation-based differential coding to reduce the maximum energy consumption, there is no obvious increase in average maximum energy consumption in the proposed algorithm when the cluster size increases.

We now study the impact of deployment range and sensing radius on the performance of the proposed scheduling algorithm. We deploy 10 camera sensors in a cluster, where the deployment range varies from 5 $\times$5 meters to 40 $\times$40 meters.

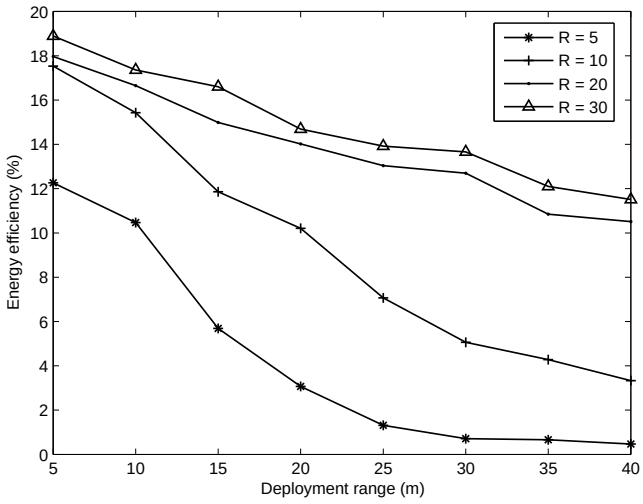

Figure 26: Energy efficiency vs. deployment range and sensing radius R.

We also vary the sensing radius R to 5, 10, 20, and 30 meters, respectively. Other parameters are the same as given above. Fig. 26 shows the impact of different deployment range and sensing radius on the energy efficiency, which is given by the percentage of reduction of the maximum energy of the approximated algorithm over the conventional TDMA-based scheduling scheme. The energy efficiency increases as the sensing radius increases, while the energy efficiency decreases as the deployment range increases. This can be attributed to the following: larger sensing radius and smaller deployment range can lead to more overlapped FoVs of the cameras and more redundancy of the observed images, so that higher energy efficiency could be achieved by differential-coding based scheduling.

The distribution of sensing directions and the offset angle of FoVs for the cameras can also affect the performance of the proposed scheduling algorithm. To evaluate these factors, we fix the other parameters in the experiment. (The cluster size is set to 10 camera sensors, the deployment range is set to 10 $\times$10 meters, and the sensing radius is $R = 30$.) We then measure the average energy efficiency under changing sensing direction distributions and offset angles. The sensing directions of each camera sensor is randomly selected within a region of degrees. The deviation

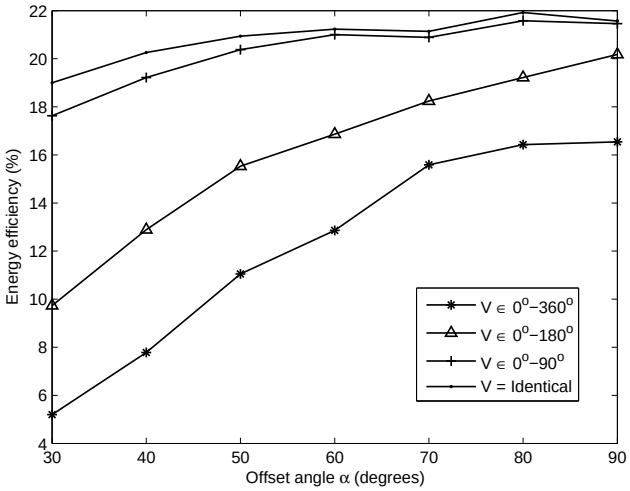

Figure 27: Energy efficiency vs. sensing direction $\vec{V}$ and offset angle α.

in the sensing directions of multiple cameras can affect the degree of correlation of the observed images. According to our previous results on correlation in Chapter 2, sensors with similar sensing directions are likely to have higher degree of correlation, resulting in more potential bit saving by differential coding. This explains the results in Fig. 27, where the lowest energy efficiency is obtained when the sensing directions are selected within 0°- 360°, while the best energy efficiency is achieved when all the cameras have identical sensing directions. As shown in Fig. 27, the energy efficiency is also related to the degree of the offset angle in the camera's FoV. The energy efficiency increases when the offset angle increases. Since a large offset angle leads to a wide FoV, there is greater probability that the cameras share large common area and have high correlation. The energy efficiency reaches the maximum value when the offset angle reaches 80°- 90°.

CHAPTER V

CORRELATION-AWARE QOS ROUTING

5.1 Introduction

WMSNs are required to provide QoS support for various applications. Many recent works have been proposed for providing QoS support at different layers of the communication stack, including QoS routing algorithms [20], QoS MAC protocols [38], and cross-layer QoS solutions [44]. In this chapter, we emphasize on the design of QoS routing protocols. Most QoS routing protocols for sensor networks are designed to support two performance metrics: *timeliness* and *reliability*. The SPEED [29] protocol achieves end-to-end soft real-time communication by maintaining a desired delivery speed across the sensor network through non-deterministic geographic forwarding. While SPEED does not consider any energy issues, a real-time power-aware routing protocol (RPAR) [11] is designed that dynamically adjusts transmission power and routing decisions to meet the packet deadlines while achieving energy efficiency. An extension of the SPEED protocol, the MMSPEED [20] provides probabilistic guarantee in the reliability domain through multipath forwarding. A node locally estimate the end-to-end reachability of a packet, and forwards multiple copies of the packet to different neighbors to reach the reliability requirements. To support the high data rate traffic of video sensors, the directional geographical routing algorithm (DGR) [10] constructs multiple disjoint paths for a video sensor by adjusting the deviation angle at each hop. High bandwidth data from a video sensor is split and forwarded through these disjoint paths. However, this algorithm is designed under the assumption that only one video sensor transmits to the sink at any time.

These existing routing solutions, however, only try to provide QoS guarantee by

properly distributing the network traffic, while the total amount of data generated by camera sensors stays the same. As a result, the existing QoS oriented approaches are still resource-demanding for WMSNs, in which large amounts of images are generated and forwarded by energy-constrained camera sensors.

To encouter this problem, collaborative multimedia in-network processing [5] is suggested to reduce the traffic volume by allowing sensor nodes to filter out uninteresting events locally or coordinate with each other to aggregate correlated data. More specifically, in wireless multimedia sensor networks, a certain degree of correlation exists among the observations at video sensors with overlapped field of views (FoVs) [15], which leads to considerable data redundancy in network traffic. It is highly desirable to remove such redundancy through effective data compression techniques.

The joint compression/aggregation and routing is an effective approach to enhance energy efficiency in sensor networks that deal with scalar data. This approach can be classified into three categories [49]: distributed source coding (DSC), routing driven compression (RDC), and compression driven routing (CDR). DSC aims to allocate the optimal coding rates to minimize the total communication cost of transporting the information collected by correlated nodes over shortest paths. In RDC, sensors send data along the preferred paths to the sink while allowing for opportunistic aggregation wherever the paths overlap. In contrast, CDR let nodes select the paths that allow for the maximum possible aggregation at each hop. In [49], the performance of routing with compression is analyzed, and a clustering scheme is designed which can provide near-optimal performance for a wide range of spatial correlations. The problem of combining tree routing and data compression with explicit side information is studied in [41]. In both [13] and [37], the problem of correlated data gathering is studied, where the goal is to minimize the total communication cost of transporting the information collected by correlated nodes. The Minimum Fusion Steiner Tree (MFST) routing algorithm [40] is proposed for energy efficient data gathering with

aggregation (fusion), in which both the data transmission cost and the cost for data fusion are optimized for applications of sensor networks. While these results work well for scalar data in sensor networks, new solutions are needed for the delivery of visual information in sensor networks. In particular, it is necessary to investigate methods to reduce the high bandwidth demand of visual information and to provide QoS support in wireless multimedia sensor networks.

We propose a correlation-aware QoS routing algorithm (CAQR) for the efficient delivery of visual information in sensor networks [16]. First, a correlation-aware inter-node differential coding scheme is developed to reduce the amount of traffic injected into the network, based on the spatial correlation of visual information studied in our previous work [15]. Then, a correlation-aware load balancing scheme is proposed to prevent network congestion by splitting the correlated flows that cannot be reduced to different paths. By integrating these correlation-aware schemes, an optimization QoS routing framework is proposed with an objective to minimize sensors' energy consumption under delay and reliability constraints. In particular, to maximize the gain of correlation-aware inter-node differential coding in the QoS routing framework, a packet delivery ratio update scheme is integrated into the routing algorithm.

To the best of our knowledge, our proposed correlation-aware QoS routing scheme is the first work that explicitly exploits the visual correlation among camera sensors to achieve energy efficient and OoS guaranteed communications in wireless multimedia sensor networks.

The remainder of this chapter is organized as follows. In Section 5.2, we review the spatial correlation characteristics of visual information for camera sensors, and discuss about video in-network compression methods. In Section 5.3, we introduce the correlation-aware QoS routing algorithm. Performance analysis and simulation results are presented in Section 5.4.

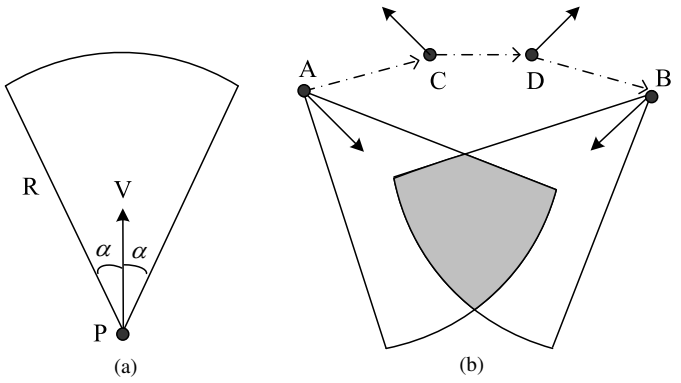

Figure 28: Camera sensors. (a) FoV (b) Overlapped FoVs.

5.2 Preliminaries

In a densely deployed WMSN, there exists correlation among the observations from video sensors with overlapped field of views (FoV). We summarize the correlation characteristics of visual information in WMSNs in this section. We introduce several metrics to quantify the degree of correlation for visual information in WMSNs. Followed by the introduction on correlation, we discuss video in-network compression mechanisms, which aim to reduce the redundancy of video streams in sensor networks.

5.2.1 Metrics for Correlation of Visual Information

A video sensor can only observe the objects within its field of view (FoV). As shown in Fig. 28(a), the FoV of a video sensor is determined by four parameters: the location of the video sensor (P), the sensing radius (R), the sensing direction ($\vec{V}$), and the offset angle (α). The sensing process of a video sensor is characterized by projection from a 3-D scene to a 2-D image, for which the key parameter is the sensor's focal length (f).

To simplify the problem, we consider the case that all the video sensors in a network are homogeneous, i.e., they have the same focal lengths (f), sensing radiuses (R), and offset angles (α). For two arbitrary video sensors V_A and V_B with FoVs F_A and F_B, suppose at a same time, their observed images are X_A and X_B, respectively.

The images X_A and X_B are correlated if F_A and F_B are overlapped with each other. In the following, we introduce two metrics for evaluating the correlation between two video sensors.

5.2.1.1 Overlapped Ratio of FoVs

The overlapped ratio of FoVs for sensors V_A and V_B, denoted by r_{AB}, is defined as the ratio of the overlapped area of the two FoVs to the area of the entire FoV of a sensor, given by

$$r_{AB} = \frac{S(F_{AB})}{S(F_A)} \tag{58}$$

where $S(F_{AB})$ ($F_{AB} = F_A \bigcap F_B$) is the overlapped area of F_A and F_B (as shown in Fig. 28(b)), and $S(F_A)$ is the area of F_A. If two video sensors have large overlapped ratio of FoVs, large portions of the two observed images are correlated. A large overlapped ratio also indicates that the two sensors are likely to observe an same event concurrently.

5.2.1.2 Spatial Correlation Coefficient

The overlapped ratio of FoVs provides a simple method to measure the correlation among multiple video sensors. To further quantitatively evaluate the correlation, we adopt the spatial correlation coefficient in Chapter 2 [15, 65]. Specifically, if two video sensors V_A and V_B can both observe an area of interest F_{AB}, a spatial correlation coefficient $\rho_{A,B}$ between V_A and V_B was derived as a function of the positions (P_A, P_B) and sensing directions $(\vec{V_A}, \vec{V_B})$ of the two video sensors as well as the overlapped FoV (F_{AB}), i.e.,

$$\rho_{A,B} = f(P_A, \vec{V_A}, P_B, \vec{V_B}, F_{AB}) \tag{59}$$

which is a normalized metric ranging from 0 to 1. In particular, a large $\rho_{A,B}$ value indicates that there is high correlation between the observations at V_A and V_B.

One promising property of this spatial correlation coefficient is its capability of

measuring the efficiency of differential coding among spatially correlated video sensors. The observations at video sensors, whether snapshot images or sequences of video frames, are large in size and have to be coded/compressed as much as possible prior to transmission. Consider two correlated images (X_A and X_B) from video sensors V_A and V_B. If each sensor compresses its observed image independently, the resulting coding rates of X_A and X_B are $R(X_A)$ and $R(X_B)$, respectively. Since X_A and X_B are correlated, X_A can be compressed using X_B as its prediction. Suppose the rate of X_A becomes $R(X_A|X_B)$ after differential coding. We define a *differential coding efficiency* as the percentage of rate saved by differential coding compared to individual coding, which is given by

$$\eta = \frac{R(X_A) - R(X_A|X_B)}{R(X_A)}. \tag{60}$$

As entropy is the lower bound for coding rate, an estimation of the differential coding efficiency can be obtained from the entropies of the image sources. Similarly, an *estimated differential coding efficiency* is defined as

$$\eta_H = \frac{H(X_A) - H(X_A|X_B)}{H(X_A)} = \frac{I(X_A; X_B)}{H(X_A)} \tag{61}$$

where $I(A; B)$ is the mutual information between X_A and X_B.

It is shown in Chapter 2 and Chapter 3 ([15, 65]) that the mutual information $I(A; B)$ is proportional to both the overlapped ratio of FoVs (r_{AB}) and the spatial correlation coefficient (ρ_{AB}). We consider the case that the individual entropies are the same for all sensor nodes ($H(X_A) = H(X_B)$), consequently, η_H is proportional to both the overlapped FoV ratio (r_{AB}) and the correlation coefficient (ρ_{AB}). The differential coding efficiency η_H will be high when both r_{AB} and ρ_{AB} are large.

5.2.1.3 Costs for estimating correlation

Both the FoV parameters and the focal lengths for video sensors could be estimated through calibration methods for distributed camera networks [18]. And this could

be done at the network deployment stage. As a result, the overlapped ratio of FoVs in (58) could be estimated without any significant costs. The spatial correlation coefficient of video sensors, which could be obtained through low computation and communications costs as shown in [15]. Therefore, the differential coding efficiency in (61) can be easily obtained.

5.2.2 Video In-network Compression

Due to the huge size of raw visual information, images and video sequences are compressed prior to transmission. A lot of standardized techniques can be applied for image and video coding, such as JPEG/JPEG 2000 for coding still images, and H.26x/MPEG standards for coding video sequences. All these standards are based on predictive coding concept. In contrast, the distributed video coding technique [23], which is based on Slepian-Wolf and Wyner-Ziv theories, allows for separate encoding of correlated sources and joint decoding at the end user. Distributed video coding (DVC) is introduced to reduce the computational complexity at the encoders, however, there is a lack of practical implementations of DVC in WMSNs. On the other hand, there are many studies on reducing the computational complexity on low-power DSPs for standardized coding techniques such as H.26x/MPEG. For these reasons, we only consider the standardized coding techniques in our work.

Standardized coding techniques can be classified into intra coding and inter coding. Intra coding refers to coding techniques that reduce the redundancy within an image, while inter coding (also called differential or predictive coding) reduces the redundancy among multiple images. Accordingly, a compressed video sequence usually consists of periodical intra coded reference frames (I frames) and inter coded frames between reference frames, including forward predicted (P) frames and bi-directional predicted (B) frames. Inter coding has much higher coding efficiency than intra coding, consequently, intra coded frames usually result in much larger sizes than inter

coded frames.

In a network of correlated video sensors, nodes can cooperate with each other and remove the redundancy among their observations. Specifically, we can perform differential coding on the intra coded (I) frames between correlated sensors. Since video sensors that are out of the communication ranges of each other can still observe a common scene [58], (i.e., they are correlated as shown in Fig. 28(b)), the differential coding of correlated sensors could be integrated in network layer operations.

Based on the discussions above, flows generated by video sensors could be classified into two categories:

1. *Intra flows*: Flows of intra coded video frames. The amount of traffic for an intra flow might be further reduced by differential coding with correlated video sensors.

2. *Inter flows*: Flows of inter coded video frames, for which the amount of traffic can hardly be further reduced.

Both types of flows need to be forwarded to the sink efficiently under QoS constraints.

5.2.3 Energy Consumption Models

The energy consumption for both video communication and processing are not negligible. Before introducing our proposed routing algorithm, we introduce the energy consumption models for video communication and compression here.

The energy consumption for transmitting and receiving l bits of data over a distance d is given as

$$E(l, d) = 2 \cdot E_{elec} \cdot l + \varepsilon_{amp} \cdot d_{hop}^{\alpha} \cdot l \tag{62}$$

where E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit, ε_{amp} is a constant for communication energy, and α is the path loss exponent.

The energy consumption for processing can be modeled as a function of supply voltage. Suppose the execution of a task consisting of N_{cyc} clock cycles, the energy consumption for processing is estimated as

$$E_{proc}(N) = N_{cyc}C_{total}V_{dd}^2 + V_{dd}(I_0 e^{\frac{V_{dd}}{nV_T}})(\frac{N_{cyc}}{f}). \quad (63)$$

The first term in (63) is the switching energy, where C_{total} is the total capacitance switched by the computation per cycle, and V_{dd} is the supply voltage. While the second term in (63) stands for the leakage energy, where f is the clock speed, and I_0, n, K, and c are processor-dependent parameters [64].

The processing burden for video in WMSNs mainly comes from the video encoding and decoding process. Fortunately, the computational complexity of standardized video codecs has been studied a lot in the literature. From the experimental results in [26] and [32], the number of clock cycles needed for encoding or decoding a video frame could be estimated. From these results together with the processor-dependent parameters in (63), we can estimate the energy consumption for encoding and decoding video frames.

5.3 Correlation-Aware QoS Routing

We propose a correlation-aware QoS routing algorithm (CAQR) for the delivery of visual information in sensor networks. By utilizing the correlation characteristics of video sensors, the algorithm achieves energy-efficient delivery of visual information in sensor networks while satisfying QoS constraints. The CAQR algorithm consists of three components: correlation groups construction, candidate node selection for correlation-aware differential coding, and QoS guaranteed next-hop selection with correlation-aware load balancing.

- *Correlation groups construction*. When the network is initially deployed, correlated video sensors are identified by grouping together the sensors that share large overlapped FoVs.

- *Intermediate node selection for correlation-aware differential coding.* For a video sensor V_A that generates an intra frame X_A, an intermediate node V_{B^*} is selected so that the redundancy in X_A is removed by performing *correlation-aware differential coding* on X_A based on the observation of V_{B^*}.

- *QoS guaranteed next-hop selection with correlation-aware load balancing.* To forward X_A to the intermediate node V_{B^*} for compression and then deliver the compressed X_A to the sink, the best next-hop node is selected, which satisfies the hop-to-hop QoS requirements while achieving minimum energy consumption. In particular, a *correlation-aware load balancing* scheme is applied to prevent network congestion by splitting the correlated flows that cannot be reduced to different paths.

5.3.1 Correlation Groups Construction

According to the analysis in Section 5.2, video sensors with large overlapped FoVs are likely to report the same event concurrently, and they are likely to have high differential coding gains. We introduce a centralized preprocessing step to cluster video sensors with large overlapped FoVs into *correlation groups*. In the following steps of the routing algorithm, correlation-aware operations are performed among video sensors that belong to the same correlation groups.

Let each video sensor report its focal length and FoV parameters to the sink. After receiving these parameters, the sink calculates the overlapped ratio of FoVs (r) (58) between any two video sensors, so that a matrix of the overlapped ratios $(r_{ij})_{N*N}$ could be obtained for a total number of N sensors in the network. We apply the hierarchical clustering algorithm [33] in the literature. By using the overlapped ratio of FoVs (r_{ij}) as a similarity metric, the hierarchical clustering algorithm iteratively groups two most similar clusters into a new cluster, and it creates a hierarchy of clusters which may be represented in a tree structure as shown in Fig. 29. Cutting

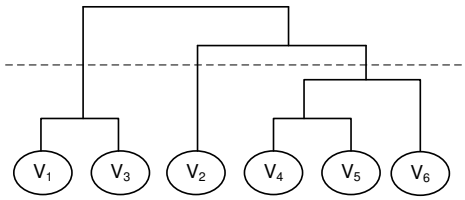

Figure 29: Correlation group construction using hierarchical clustering.

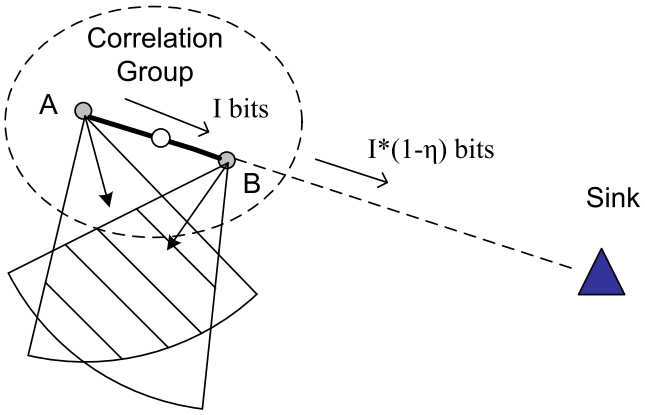

Figure 30: Correlation-aware differential coding.

the hierarchical tree at a given similarity level results in a group of clusters, e.g., clusters $\{V_1,V_3\}$, $\{V_2\}$, and $\{V_4, V_5, V_6\}$ in Fig. 29. The readers are referred to [33] for more details of the hierarchical clustering algorithm.

After running the clustering algorithm, the sink broadcasts the results of clustering and assigns a group ID for each group. Each video sensor will be notified which correlation group it belongs to and other video sensors' sensing parameters in the same correlation group. The correlation groups only need to be constructed once during the deployment of a WMSN, and the whole algorithm is run at the powerful sink. Therefore, this procedure is applicable in wireless multimedia sensor networks.

5.3.2 Intermediate Node Selection

The traffic of intra flows from a video sensor could be further reduced through differential coding with other correlated sensors. We introduce a correlation-aware inter-node differential coding scheme for the routing of intra flows.

As shown in the example in Fig. 30, a video sensor V_A needs to find a route for its intra frame X_A to the sink. It could find another candidate sensor in the same group that is closer to the sink to perform differential coding. Suppose sensor V_B is in the same group as V_A and its distance to the sink is closer than V_A ($d_{BS} < d_{AS}$). From our correlation model, we can estimate the differential coding efficiency between V_A and V_B, η_{AB} from (61). If the size of the intra frame X_A is I_A, we can estimate the saved bits from differential coding as $I_A \cdot \eta$. We introduce an energy gain to evaluate the potential energy efficiency of differential coding between nodes V_A and V_B:

$$G_E(V_A, V_B) = \frac{E_{com}\{I_A \cdot \eta_{AB}\}}{E_{proc}\{I_A\}}. \tag{64}$$

The numerator in the gain function is the communication energy for the bits that are saved from differential coding. It stands for the benefits brought by differential coding. This communication energy is not only related to the number of saved bits, but also related to the distance and number of hops from sensor V_B to the sink. We can estimate E_{com} using the estimated number of hops from node V_B to the sink ($\hat{N}_B^{hops}$) and the average one-hop distance (d_{hop}), given by

$$E_{com}\{I_A \cdot \eta_{AB}\} = \hat{N}_B^{hops} \cdot E(I_A \cdot \eta_{AB}, d_{hop}) \tag{65}$$

where $E(I_A \cdot \eta_{AB}, d_{hop})$ is obtained from (62) and $\hat{N}_B^{hops}$ can be estimated by

$$\hat{N}_B^{hops} = \max\left(\lceil \frac{d_{BS}}{d_{hop}} \rceil, 1\right). \tag{66}$$

The denominator in (64) is the energy costs for performing differential coding, i.e., the processing energy of differential coding at sensor V_B, including the decoding of the intra frame and the differential coding of the intra frame X_A with respect to the frame X_B at node V_B. The energy for processing is related to video frame size and video processing hardware. This term could be estimated from equation (63) using the parameters in [26] and [32].

For the routing of an intra frame generated at node V_A in correlation group $\mathcal{G}(V_A)$, we aim to find the best intermediate node in the same group with an objective to maximize the energy gain of differential coding. This problem can be formulated as follows.

Differential coding-based intermediate node selection (DCIS) problem

$$Find: \quad V_{B^*} = arg \max_{V_B \in \mathcal{G}(V_A)} G_E(V_A, V_B) \tag{67}$$

$$Subject\ to: \quad d_{BS} < d_{AS},\ G_E(V_A, V_B) > 1. \tag{68}$$

A node V_B for differential coding should satisfy two conditions: i) V_B is closer to the sink than V_A, which is given as $d_{BS} < d_{AS}$ above; ii) the energy gain for the differential coding is larger than 1. Among all the nodes in group $\mathcal{G}(V_A)$ that satisfy these two conditions, the node that generates the maximum energy gain (V_{B^*}) is selected as the candidate node for differential coding.

After node V_A determines node V_{B^*} for differential coding, it sends a request message to V_{B^*}, and V_{B^*} will send back a reply message. In this way, V_{B^*} becomes an intermediate destination for the intra frames generated by V_A. The routing of intra frames from V_A to the sink can then be split into two steps: the intra frames from V_A will be forwarded to V_{B^*} first; V_{B^*} will further compress the frame and then forward it to the sink. In both steps the routes will be chosen to minimize energy consumption subject to given QoS and load balancing constraints, which will be explained in Section 5.3.3.

5.3.3 Correlation-aware QoS Routing

We now introduce an integrated QoS routing algorithm for the delivery of visual information. It allows each node to distributively select the optimal next hop with the objective of minimizing the energy consumption and satisfying the QoS requirements in the delay and reliability domain.

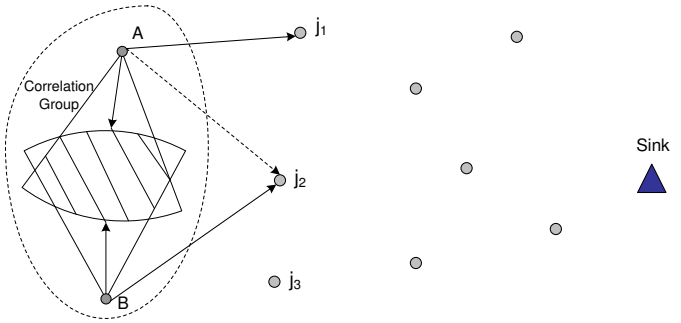

Figure 31: Correlation-aware load balancing.

Suppose a node i needs to forward a video flow to the destination N. (If the flow needs correlation-aware differential coding, the destination N will be an intermediate node for differential coding or the sink. In other cases the destination N corresponds to the sink.) We define the forwarding neighbor set of node i as a set of its neighbors that are closer to the sink than itself, denoted by $\mathcal{F}_i$. The next hop node is selected from $\mathcal{F}_i$ according to the following rules.

Distributed correlation-aware QoS routing (DCR) problem

$$Given: \quad i, N, j \in \mathcal{F}_i, V_{id}, \mathcal{G}(V_{id}), \{R_0^C, \ldots, R_N^C\}$$

$$Find: \quad j^*, R_{ij^*}^C \tag{69}$$

$$Minimize: \quad E(L/R_{ij}^C, d_{ij}) \tag{70}$$

$$Subject\ to: \quad \frac{L}{R \cdot R_{ij}^C} + \overline{t_{ij}^q} < T_{ij} \tag{71}$$

$$\frac{\gamma}{1-\gamma}(\Delta t_{ij}^q)^2 \leq (T_{ij} - \frac{L}{R \cdot R_{ij}^C} - \overline{t_{ij}^q})^2 \tag{72}$$

$$pr_{ij} \geq PR_{ij} \tag{73}$$

$$\sum_{v^j \in \mathcal{L}\{v^j\}} x_{v^j} \leq w \tag{74}$$

The locally optimal next hop j^* is the node that results in the minimum energy consumption under local delay, local reliability, and correlation-aware load balancing requirements. Apart from the optimal next hop, the algorithm also determines a proper channel coding rate for the link from i to j^*, $R_{ij^*}^C$, from a set of predefined channel coding rates $\{R_0^C, \ldots, R_N^C\}$.

The objective in (70) is to minimize energy consumption. From the energy estimation equation in (62), we can easily obtain the energy consumption for transmitting a packet of L bits data and header with channel coding rate R_{ij}^C as $E(L/R_{ij}^C, d_{ij})$, where d_{ij} is the hop distance.

In the above problem, equations (71) and (72) are the local delay requirements, (73) is the local reliability requirement, and (74) is the constraint for correlation-aware load balancing. We explain each constraint in the following sections.

5.3.3.1 ***Local delay requirements***

We use a geographic based mechanism to map end-to-end delay requirements to local delay requirements. Suppose a video flow v at node i needs to be delivered to the destination N within time T_{iN}. The local delay constraint, T_{ij}, is given as

$$T_{ij} = \left(\frac{d_{iN} - d_{jN}}{d_{iN}} \right) \cdot T_{iN} \tag{75}$$

where d_{iN} is the distance from node i to the destination, and d_{jN} is the distance from node j to the destination.

The delay of a hop is related to the underlying MAC mechanisms. We consider a contention-free MAC in our context. Under this assumption, the delay of a hop mainly consists of the transmission delay and the queueing delay. The transmission delay for a packet from node i to node j can be calculated as $\frac{L}{R \cdot R_{ij}^C}$, where L is the length of the packet, R is the transmission rate, and R_{ij}^C the channel coding rate. We denote the queueing delay from node i to j by t_{ij}^q. Then the total delay from node i to j is given by $\frac{L}{R \cdot R_{ij}^C} + t_{ij}^q$.

We provide probabilistic guarantee for one-hop delay, in which the probability that a packet is delivered within deadline should not be below γ. This constraint is given by

$$P\left(\frac{L}{R \cdot R_{ij}^C} + t_{ij}^q \leq T_{ij} \right) \geq \gamma. \tag{76}$$

It can be expressed in an alternative way as

$$P\left(\frac{L}{R\cdot R_{ij}^{C}}+t_{ij}^{q}\geq T_{ij}\right)\leq 1-\gamma. \tag{77}$$

We let node j maintain the delays of packets in a recent period, and from the delay statistics, we can estimate the average queueing delay $\overline{t_{ij}^{q}}$, and the variance of queueing delay $(\Delta t_{ij}^{q})^2$. As a result, the average single hop delay is $\frac{L}{R\cdot R_{ij}^{C}}+\overline{t_{ij}^{q}}$, while the variance of single hop delay is $(\Delta t_{ij}^{q})^2$.

According to one-sided Chebyshev's inequality, for a random variable X with mean μ and variance σ^2, it satisfies

$$P(X-\mu\geq k)\leq\frac{\sigma^2}{\sigma^2+k^2}, k>0. \tag{78}$$

By applying the Chebyshev's inequality on the left part of (77), we find

$$P\left(\frac{L}{R\cdot R_{ij}^{C}}+t_{ij}^{q}\geq T_{ij}\right)\leq\frac{(\Delta t_{ij}^{q})^2}{(\Delta t_{ij}^{q})^2+(T_{ij}-\frac{L}{R\cdot R_{ij}}-\overline{t_{ij}^{q}})^2} \tag{79}$$

and

$$T_{ij}-\left(\frac{L}{R\cdot R_{ij}^{C}}+\overline{t_{ij}^{q}}\right)>0. \tag{80}$$

Based on (79) and (80), we have derived two constraints to satisfy the probabilistic delay guarantee in (77), which are given in (71) and (72). The condition (80) corrresponds to the constraint (71). Furthermore, comparing (79) and (77), if the condition

$$\frac{(\Delta t_{ij}^{q})^2}{(\Delta t_{ij}^{q})^2+(T_{ij}-\frac{L}{R\cdot R_{ij}}-t_{ij}^{q})^2}\leq 1-\gamma \tag{81}$$

is met, the probabilistic delay guarantee inequation (77) could be satisfied. From this condition we can obtain the constraint (72) in the routing problem.

5.3.3.2 ***Local reliability requirements***

Packet loss at a wireless link is mainly caused by channel errors and packet drops for congestion control. To combat packet loss caused by channel errors, we incorporate a

dynamic channel coding scheme in the routing algorithm to adapt to varying channel conditions. Apart from selecting the next hop for transmission, the routing algorithm also selects a proper channel coding rate. The channel coding rate for link i to j, R_{ij}^C, is chosen from a set of predefined channel coding rates $\{R_0^C, \ldots, R_N^C\}$, where $R_0^C > \ldots > R_N^C$. A smaller channel coding rate indicates more redundancy being added to a packet and better error resilience performance. Based on the detected link SINR, the channel coding rate with the minimum redundancy is selected from the candidate rates that meet the QoS constraints.

For evaluating reliability, we introduce a measure called *packet delivery ratio*, the percentage of packets successfully delivered to the destination. If we require that each hop on a route should provide the same level reliability, the required packet delivery ratio from node i to node j, PR_{ij}, can be estimated as

$$PR_{ij} = PR^{1/\hat{N}_{ij}} \tag{82}$$

where PR is the required packet delivery ratio given by the applications, and $\hat{N}_{ij}$ is the estimated number hops from i to the destination if j is selected as its next hop, i.e.,

$$\hat{N}_{ij} = \max\left(\lceil \frac{d_{iN}}{\hat{d_{ij}}} \rceil, 1\right) \tag{83}$$

where $\hat{d_{ij}}$ is the projection of d_{ij} onto the line connecting node i with the sink. Next we explain how to obtain the value of PR and how to set PR for correlation-aware differential coding.

An end-to-end video application usually cares if a video frame can be successfully decoded or not. Therefore, we use the *probability that a video frame is successfully decoded* [77] as a metric to evaluate the reliability of frame delivery. We denote this probability by P_D. Suppose a video frame X is packed into n packets for transmission. It will be decodable only when enough of the n packets are received correctly. We introduce the parameter *frame decodable threshold* [77], denoted by DT, to represent

the percentage of packets needed to decode a frame. This threshold is dependent on specific video coders and their error recovering capabilities. Let PR be the packet delivery ratio of each packet. The probability that at least DT percent of the packets of the frame are successfully delivered, denoted by $\varphi(X)$, is estimated from a function of n, DT, and PR, given by

$$\varphi(X) = \varphi(n, DT, PR) = \sum_{i=\lceil n \cdot DT \rceil}^{n} \binom{n}{i} \cdot PR^i \cdot (1 - PR)^{n-i}. \tag{84}$$

An intra coded frame is decodable if at least DT percent of the packets are delivered to the sink. For example, if a video sensor V_A has generated an intra frame X_A. The probability that X_A is successfully decoded is given as

$$P_D(X_A) = \varphi(X_A) = \varphi(n_A, DT, PR_A) \tag{85}$$

where n_A is the number of packets for X_A and PR_A is the packet delivery ratio for each packet.

In our problem, given a required $P_D(X_A)$ from an application, the number of packets for X_A (n_A), and the frame decodable threshold (DT), the required packet delivery ratio (PR_A) is estimated and assigned to each packet for the QoS routing algorithm.

After correlation-aware differential coding is performed for an intra video frame, it becomes an inter frame, resulting in reduced packets but more dependency among frames. Consider the differential coding of frame X_A using the prediction of frame X_B. The intra frame X_A becomes inter frame X'_A after differential coding. Suppose the number of packets in X'_A is reduced from n_A to n'_A in this process. To decode frame X'_A at the end user, DT percent of the n'_A packets needs to be successfully decoded. More importantly, its reference frame X_B is also required to be successfully decoded. Therefore, the probability that frame X'_A is decodable is given as

$$\begin{aligned} P_D(X'_A) &= P_D(X_B) \cdot \varphi(X'_A) = \varphi(X_B) \cdot \varphi(X'_A) \\ &= \varphi(n_B, DT, PR_B) \cdot \varphi(n'_A, DT, PR'_A) \end{aligned} \tag{86}$$

where n_B and n'_A are the number of packets for X_B and X'_A, respectively, and PR_B and PR'_A are their corresponding packet delivery ratios.

To maintain the quality of video frames, the decodable probability of a frame after correlation-aware differential coding has to be consistent with that before correlation-aware differential coding. As the decodable probability of a frame is related to the packet delivery ratio (PR) in (84), we need to update the required packet delivery ratio (PR) when correlation-aware differential coding is performed. We formulate a problem to update the required packet delivery ratio (PR) for correlation-aware coding as follows.

Packet delivery ratio update (PDRU) problem

$$\begin{aligned} Given: \quad & n_A, n_B, n'_A, DT, P_D^{req}(X_A), P_D^{req}(X_B), p_b \\ Find: \quad & PR_A^{new}, PR_B^{new} && (87) \\ Maximize: \quad & E\{\eta_c\} && (88) \\ Subject\ to: \quad & \end{aligned}$$

$$\varphi(n_B, DT, PR_B^{new}) \cdot \varphi(n'_A, DT, PR_A^{new}) \geq P_D^{req}(X_A) \quad (89)$$

$$\varphi(n_B, DT, PR_B^{new}) \geq P_D^{req}(X_B) \quad (90)$$

We consider the case without correlation-aware coding first. For the two intra frames X_A and X_B, suppose their required frame decodable probabilities are $P_D^{req}(X_A)$ and $P_D^{req}(X_B)$, which are assigned by applications. Given the number of packets in these two frames, n_A and n_B, and the frame decodable threshold (DT), based on (85), we can determine the required packet delivery ratios for these two frames, denoted by PR_A^{old} and PR_B^{old}, to satisfy the required $P_D^{req}(X_A)$ and $P_D^{req}(X_B)$.

After correlation-aware coding, the two frames become the inter frame X'_A and the intra frame X_B. Suppose the required packet delivery ratios for X'_A and X_B are PR_A^{new} and PR_B^{new}, respectively. Based on (85) and (86), the frame decodable probability of X_B is $\varphi(n_B, DT, PR_B^{new})$, and the frame decodable probability of X'_A is given by

$\varphi(n_B, DT, PR_B^{new}) \cdot \varphi(n'_A, DT, PR_A^{new})$. The resulting frame decodable probabilities of these two frames should also meet the application requirements, which are given as constraints in (89) and (90).

As introduced above, the proposed routing algorithm incorporates a dynamic channel coding scheme, where the channel coding rate is selected based on reliability requirements and channel condition. Therefore, if the value of required packet delivery ratio is changed, the channel coding rate might need to be updated, which may further influence the amount of traffic in the network. Taking into account the effect of channel coding, we introduce a metric called *differential coding efficiency after channel coding* as

$$\eta_c = \frac{(\frac{n_B \cdot L}{R^C_{B\_old}} + \frac{n_A \cdot L}{R^C_{A\_old}}) - (\frac{n_B \cdot L}{R^C_{B\_new}} + \frac{n'_A \cdot L}{R^C_{A\_new}})}{\frac{n_A \cdot L}{R^C_{A\_old}}} \tag{91}$$

where $R^C_{A\_old}$ and $R^C_{B\_old}$ are the channel coding rates of X_A and X_B without correlation-aware coding, $R^C_{A\_new}$ and $R^C_{B\_new}$ are the channel coding rates of X'_A and X_B after correlation-aware coding, and L is the packet length. With a similar form as the definition in (60), this metric describes the percentage of saved bits for correlation-aware coding considering dynamic channel coding.

To meet the required packet delivery ratio, the channel coding rate (R^C) is chosen based on specific channel conditions (the received SNR and the corresponding bit error rate). From the many possible solutions of PR_A^{new} and PR_B^{new} that meet the constraints in (89) and (90), we would like to find the ones that maximize the gain of correlation-aware coding under varying channel conditions. This objective is given in (88) as the maximization term $E\{\eta_c\}$, the average value of η_c for a range of possible SNRs.

The packet delivery ratio update problem can be solved by enumerating possible combinations of PR_A^{new} and PR_B^{new} values and finding the best combination that maximizes the average gain, $E\{\eta_c\}$. First, based on the constraints in (89) and (90),

we find out the possible combinations of PR_A^{new} and PR_B^{new} values that satisfy the constraints. For each possible combination of PR_A^{new} and PR_B^{new}, we estimate the corresponding channel coding rates, $R_{A\_new}^C$ and $R_{B\_new}^C$. We take the estimation of $R_{A\_new}^C$ as an example. If given a certain bit error rate, from a set of predefined channel coding rates $\{R_0^C, \ldots, R_N^C\}$, we select the largest channel coding rate that satisfies the required packet delivery ratio (PR_A^{new}). After the channel coding rates $R_{A\_new}^C$ and $R_{B\_new}^C$ are estimated, the differential coding efficiency after channel coding in (91) could be determined for this specific bit error rate. We assume that the distribution of possible received SNRs and the corresponding bit error rates are known in advance, so that the average gain $E\{\eta_c\}$ can be calculated based on the distribution. The solution to this problem will be the required packet delivery ratios PR_A^{new} and PR_B^{new} that result in the largest average gain $E\{\eta_c\}$. These solutions are then used in constraint (73) in the correlation-aware QoS algorithm.

5.3.3.3 ***Correlation-aware load balancing***

The differential coding scheme introduced above can reduce the amount of traffic in the network if video sensors in a correlation group have high differential coding gains. For flows from the same correlation group that cannot be further compressed, the presence of traffic congestion becomes evident in that video sensors from the same correlation group tend to report the same event and generate traffic concurrently. To solve this problem, we introduce a correlation-aware load balancing operation. The basic idea is to split these flows to different paths so that the probability of network congestion could be reduced. As shown in Fig. 31, two video sensors in a correlation group, V_A and V_B share large overlapped FoVs, however, the differential coding gain is low according to our correlation model. As they are likely to generate large amounts of traffic concurrently, we can try to split the video flows from the two sensors to different paths.

To achieve correlation-aware load balancing, each node keeps a list of source nodes and their corresponding group IDs that it has generated or routed in a recent period. Supppose node i wants to find a next hop for a flow generated by node V_{id} at correlation group $\mathcal{G}(V_{id})$. Its candidate neighbor, node j, has a list of source nodes of the flows that it has routed in a recent time period, denoted by ($\mathcal{L}\{v^j\}$). In addition, each source node v^j in the list is associated with its correlation group ID $\mathcal{G}(v^j)$. Node j periodically exchanges this list with its neighbors, so that node i is aware of it. For the current flow which is generated by V_{id}, node i can check if the candidate j has routed flows for other nodes in the same correlation group.

We define a variable x_{v^j} to indicate if a source node v^j is in the same group as V_{id}, which is given by,

$$x_{v^j} = \begin{cases} 1, & \text{if } \mathcal{G}(v^j) = \mathcal{G}(V_{id}) \text{ and } v^j \neq V_{id} \\ 0, & \text{otherwise} \end{cases} \tag{92}$$

The number of nodes in list $\mathcal{L}\{v^j\}$ that are in the same group as V_{id} can be expressed as $\sum_{v^j \in \mathcal{L}\{v^j\}} x_{v^j}$. For the load balancing purpose, the algorithm should prefer to choose a next hop node with a smaller $\sum_{v^j \in \mathcal{L}\{v^j\}} x_{v^j}$, which, as indicated in constraint (74), cannot exceed a threshold value w. In this way, for flows from the same correlation groups that cannot be further compressed, we can penalize the case that they share the same forwarding node concurrently (e.g., node j_2 in Fig. 31), thereby reducing the possibility of congestion.

5.3.4 Protocol Operation

The proposed CAQR routing algorithm is summarized in *Algorithm 10*. When a wireless multimedia sensor network is deployed, correlation groups are first constructed. After that, if a sensor V_A has a video frame X_A to transmit, it encounters two scenarios: i) If X_A is an inter frame, V_A directly send X_A to the sink node through multihop communications, where the next hops are selected by performing *Algorithm 11* that

solves the DCR problem in Section 5.3.3. ii) If X_A is an intra frame, V_A selects the optimal intermediate node V_{B^*} by solving the DCIS problem in Section 5.3.2. The QoS constraints for frames X_A and X_{B^*}, which are generated by V_A and V_{B^*}, respectively, are set by solving the PDRU problem in Section 5.3.3.2. Otherwise, if no such intermediate node can be found, the sensor V_A directly send X_A to the sink node. In both cases, *Algorithm 11* is executed to find the optimal next hop nodes.

Algorithm 11 is performed as follows. First, each sensor can find out all the next hop candidate nodes that satisfy the load balancing constraint in (74). For each candidate node, the sensor finds out the largest channel coding rate R_{ij}^C from $\{R_0^C, \ldots, R_N^C\}$ such that the reliability constraint (73) can be satisfied. Using this largest channel coding rate, the sensor checks if the local delay constraints (71) and (72) are met. If so, this candidate node satisfies all the constraints, and the corresponding energy consumption in (70), $E(L/R_{ij}^C, d_{ij})$ can be obtained. Then, the candidate node that results in the smallest energy consumption $E(L/R_{ij}^C, d_{ij})$ is selected as the next hop node. In cases that no candidate nodes can satisfy all the four constraints, the load balancing constraint (74) could be relaxed by increasing the threshold w by a small amount, so that more nodes in the neighbor set could be considered as next hop candidates.

5.4 Performance Evaluation

As the proposed routing algorithm relies on the accurate prediction of the differential coding efficiency of practical codecs, we first evaluate the validity of the proposed differential coding efficiency estimation method. Then, we study the gain of correlation-aware differential coding in the QoS routing algorithm. After these preliminary evaluations, we test the networking performance of the proposed routing algorithm in terms of delay, frame delivery ratio, and energy consumption.

Algorithm 10 Correlation-Aware QoS Routing Overview

Construct correlation groups as shown in Section 5.3.1.
while Sensor V_A has frames to transmit **do**
  Get an image frame X_A at V_A.
  if $Type(X_A) = intra$ **then**
    if A node V_{B^*} for differential coding is found by solving the DCIS problem **then**
      Set the QoS constraints for X_A and X_{B^*} by solving the PDRU problem.
      Send X_A from V_A to V_{B^*} by *Algorithm 11*.
      Perform differential coding on X_A from the prediction of X_{B^*}.
      Send X_A from V_{B^*} to the sink by *Algorithm 11*.
    else
      Send X_A to the sink by *Algorithm 11*.
    end if
  else
    Send X_A to the sink by *Algorithm 11*.
  end if
end while

Algorithm 11 Energy Efficient and QoS Guaranteed Next-hop Selection

$C_i = \{j | \sum_{v^j \in \mathcal{L}\{v^j\}} x_{v^j} \leq w, j \in \mathcal{F}_i\}$
for $j \in C_i$ **do**
  $R = \{R_{ij}^C | pr_{ij} \geq PR_{ij}, R_{ij}^C \in \{R_0^C, \ldots, R_N^C\}\}$
  if $R \neq \emptyset$ **then**
    Find $\{R_{ij}^{C*}\} = \max\{R_{ij}^C \in R\}$
    $s.t.\ \frac{L}{R \cdot R_{ij}^C} + \overline{t_{ij}^q} < T_{ij}$
    $\frac{\gamma}{1-\gamma}(\Delta t_{ij}^q)^2 \leq (T_{ij} - \frac{L}{R \cdot R_{ij}^C} - \overline{t_{ij}^q})^2$
  end if
end for
$j^* = \arg\min_{j \in C_i} E(L/R_{ij}^C, d_{ij})$

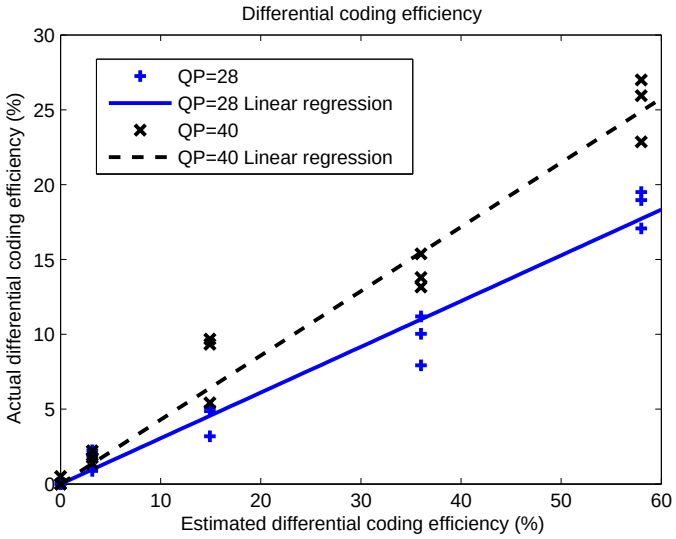

Figure 32: Estimation of differential coding efficiency.

5.4.1 Coding Efficiency Prediction

The correlation and entropy-based framework in Section 5.2 provides a simple way to estimate the differential coding efficiency between video sensors. To verify the performance of the proposed method, we compare the estimated coding efficiency to the actual coding efficiency from practical coding experiments.

We deploy two cameras in a field and record their FoV parameters. The cameras' sensing radius is 30 meters and the offset angle is 60 degrees. By varying the locations and sensing directions of the two cameras, different degrees of correlation can be obtained, resulting in different values of the estimated coding efficiency (η_H) in (61). At each deployment, we let each camera capture one image, and the actual differential coding efficiency (η) in (60) is calculated from the coding rates by performing differential coding using the H.264 standard coding algorithm with reference software JMVC 2.5 [3].

Comparisons of the measured η_H and η values are shown in Fig. 32. In the experiment, 5 values of estimated differential coding efficiency η_H are obtained, and for each η_H, 3 different groups of images are used for differential coding. The coding

rates of differential coding is obtained under two quantization steps (QP= 28 and 40), where a small quantization step corresponds to better image quality and larger coding rate.

According to the data points in Fig. 32, if given the same prediction of η_H, we find that a larger quantization step results in larger values of actual coding efficiency. Since larger quantization steps allow for more distortion, they may have more bit savings for differential coding. When the quantization step is fixed, the actual coding efficiency (η) is approximately proportional to the predicted coding efficiency (η_H). Therefore, the actual differential coding efficiency η can be predicted by a linear function of η_H, given by

$$\eta = k \cdot \eta_H \tag{93}$$

where k is a ratio that depends on the performance of specific video encoders as well as the encoder parameters (e.g., quantization step). By performing linear regression on the data points in Fig. 32, we find that $k = 0.31$ for $QP = 28$ and $k = 0.43$ for $QP = 40$. Based on the value of k, we find that the average absolute error for this prediction method is 0.01 and the worst case error is 0.03. This linear relationship between the predicted results and experimental performance validates the applicability of the proposed coding efficiency prediction method.

5.4.2 Coding Efficiency in QoS Routing

In this section, we evaluate the gain of correlation-aware coding when it is implemented in the QoS routing algorithm. We find solutions to the packet delivery ratio update problem in Section 5.3.3, and based on the updated required packet delivery ratio, we test the best average differential coding efficiency after channel coding in (91).

The parameters in the packet delivery ratio update problem (88) are determined as follows. The average size of an intra frame is determined from the statistics of

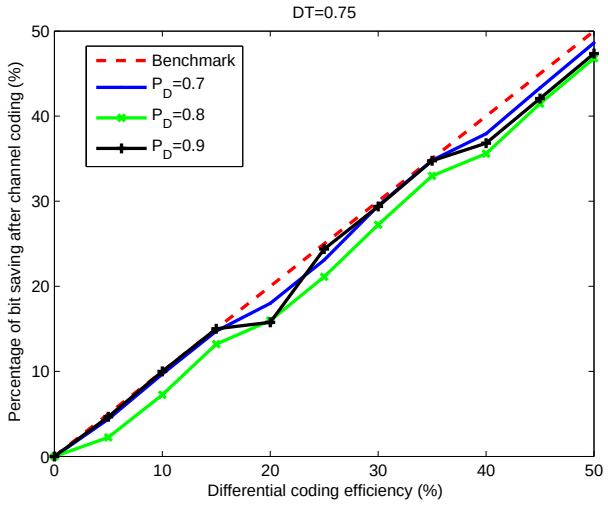

Figure 33: Differential coding efficiency after channel coding (DT=0.75).

the video traces in [55]. The payload length of a packet is set to 50 Bytes. The number of packets of intra frames X_A and X_B, n_A and n_B, can be estimated from the average size of the frame and the packet length. As for the number of packets of the coded inter frame X'_A, n'_A, it is calculated from n_A and the differential coding efficiency η, which is given by $n'_A = n_A \cdot (1 - \eta)$. We use a series of block codes with structures (n, k, t) [36] for dynamic channel coding. The block length is set to 127, and the number of correctable bits t varies from 1 to 31. A single hop scenario with BPSK modulation is considered, where the received SNR is assumed to be uniformly distributed between -5 dB and 15 dB.

The required frame decodable probability (P_D) is assigned by specific applications. We assume that the required P_D for X_A and X_B are the same, and use three different P_D values for the test: 0.7, 0.8, and 0.9. The frame decodable threshold DT is related to the error recovering capability of video decoders. Here DT is set to two values: 0.75 and 0.9. Let the differential coding efficiency η (without channel coding) vary from 0 to 0.5. For each combination of P_D, DT, and η, we solve for the best required packet delivery ratio (PR) for correlation-aware coding, so that the average differential coding efficiency after channel coding ($E\{\eta_c\}$) can be obtained.

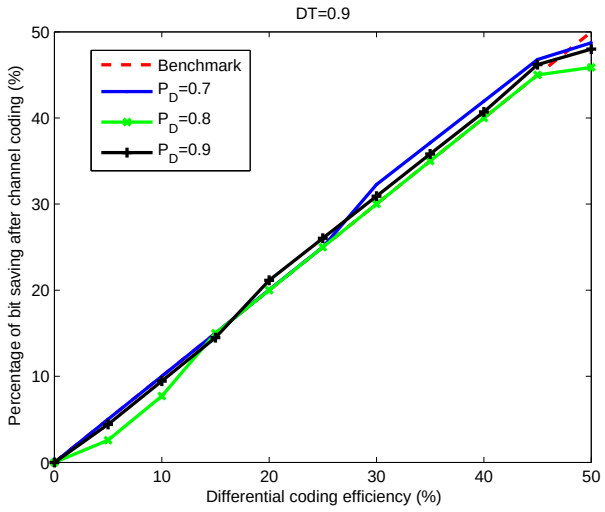

Figure 34: Differential coding efficiency after channel coding (DT=0.9).

Fig. 33 and Fig. 34 show the results of $E\{\eta_c\}$ as a function of η. In both figures, a dotted line is plotted as a benchmark line that corresponds to $E\{\eta_c\} = \eta$. This line can represent the case of error-free channels: if no channel coding is needed, η_c is always equal to η. The other lines in Fig. 33 and Fig. 34 show the average differential coding efficiency in lossy channel conditions. For different combinations of P_D and DT values, the average differential coding efficiency after channel coding is close to the benchmark line. It indicates that by properly updating the required packet delivery ratio, correlation-aware coding can still reduce the traffic load in the network. Since there is only a little fluctuation of the lossy-channel case compared to the benchmark line, the efficiency of differential coding after channel coding (η_c) could still be approximated by the original differential coding efficiency (η).

5.4.3 Correlation-Aware QoS Routing Algorithm

The performance of the proposed CAQR algorithm is then evaluated using a distributed network simulator in Java. In a field of 100m ×100m, 49 video sensors are deployed in a grid structure, and a sink node is placed in a corner of the field. The

sensing directions of the video sensors are uniformly chosen so as to ensure full coverage of the field. From the sensing parameters of the video sensors in Table I, we can obtain the correlation parameters, i.e., the ratio of overlapped FoVs (58) and the estimated differential coding efficiency (93). After these parameters are calculated, video sensors are clustered into correlation groups as discussed in Section 5.3.1.

The traffic for the video sensors is generated based on the features of video surveillance and environmental monitoring applications. Specifically, we place a target within the field and let it move around according to the Random Waypoint Mobility model where the pause time is set to 0. In this way a sequence of events can be generated, where each event is tagged by the location of the target and the corresponding timestamp. A video sensor is triggered to capture an image when it detects the event in its FoV. By launching the target from 10 different locations, we can generate 10 sequences of events representing different network traffic scenarios. In addition, the size of a captured video frame is in QCIF format with resolution 176×144, while the size of an encoded video frame is simulated based on the video traces provided in [55]. If correlation-aware differential coding is performed, the size of a frame is updated based on the actual coding efficiency, which is obtained from (93).

Other key parameters for the simulation of the proposed algorithm are given as follows. The bandwidth of the channel is set to 1 Mb/s, and the transmission range is set to 15 meters. For the MAC layer, we use the TDMA scheduling algorithm for sensor networks in [19], and the length of a time slot is set to 20 ms. The parameters for the communication energy consumption model in (62) are $E_{elec} = 50$ nJ/b, $\varepsilon = 10$ pJ/b/m^2, and $\alpha = 2$. As for the energy consumption for processing in (63), the parameters in [64] and [56] are used, which are given as follows: $C_{total} = 0.67$ nF, $I_0 = 1.196$ mA, $V_T = 26$ mV, $K = 239.28$ MHz/V, and $c = 0.5$ V. To evaluate the costs for performing differential coding between correlated sensors in (64), we need

Table 6: Parameters for correlation-aware QoS routing

Offset angle	60	Image size	176×144
Sensing radius	30	Intra period	2
Transmission rate	1 Mbps	DT	0.8

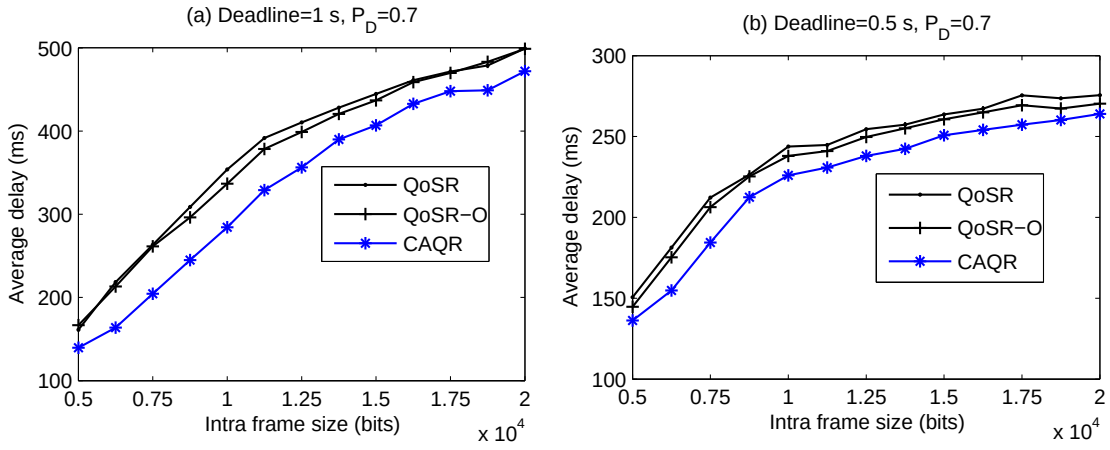

Figure 35: Average delay for correlation-aware QoS routing.

to know the number of clock cycles (N_{cyc}) for processing a video frame. From the experimental results in [26] and [32], we have estimated that for QCIF sequences, the number of clock cycles needed for encoding a frame is 2.3 Mcycles, while the number of clock cycles needed for decoding a frame is 0.14 Mcycles.

We evaluate the performance of the CAQR algorithm under varying traffic load and different QoS requirements. For comparison, we design two other relevant routing algorithms: QoSR and QoSR-O. The QoSR algorithm is the QoS routing algorithm in Section 5.3.3 without any correlation-aware operations. The QoSR-O algorithm is the QoSR algorithm added with opportunistic compression: video sensors send their data along paths chosen by the QoSR algorithm, and in the routing process, if a relay node finds that a frame is eligible for differential coding based on (64), correlation-aware differential coding is performed.

In the simulation we change the amount of traffic injected in the network by adjusting the source coding rates of the video frames. More specifically, the source

coding rates are adjusted by varying the quantization steps (QP) of video coding. A larger quantization step corresponds to more distortion in an image, resulting in a smaller coding rate. Based on the video traces for QCIF frames in [55], when QP changes from 28 to 40, we have estimated that the size of an encoded I frame ranges from around 2 $\times 10^4$ bits to 0.5 $\times 10^4$ bits. We also found from the traces [55] that in average the size of an encoded P frame is around 0.2 times that of an encoded I frame. These statistics are used to set the coding rates in the simulation. For each rate setting, experiments on the aforementioned 10 different sequences of events are launched, and we measure the average performance of the 10 sequences of events.

Fig. 35 shows the average delay under different source coding rates, where in Fig. 35 (a) the deadline is set to 1 second and in Fig. 35 (b) the deadline is set to 0.5 second. In both cases we only consider the delay of packets that are received within the deadline. It can be found that the average delay increases as the source coding rate increases. Although the QoSR-O algorithm performs opportunistic differential coding along routing paths, it does not bring much performance enhancement compared to the QoSR algorithm. The proposed CAQR algorithm fully exploits the correlation in wireless multimedia sensor networks, and reduces the transmission of redundant information as much as possible. Therefore, it is seen in Fig. 35 that the proposed algorithm results in less average delay compared with the QoSR and the QoSR-O.

Next we evaluate the energy consumption of the proposed routing algorithm. The total energy consumption consists of the communication energy for sending and receiving packets and the energy for processing the video frames. If given the same event, the processing energy for sensing video frames and the energy for encoding local video frames will be the same for CAQR, QoSR, and QoSR-O, whereas differential coding along routing paths will introduce extra processing energy for CAQR and QoSR-O. Therefore, when calculating the total energy consumption, we just consider the communication energy and the processing energy for differential coding along

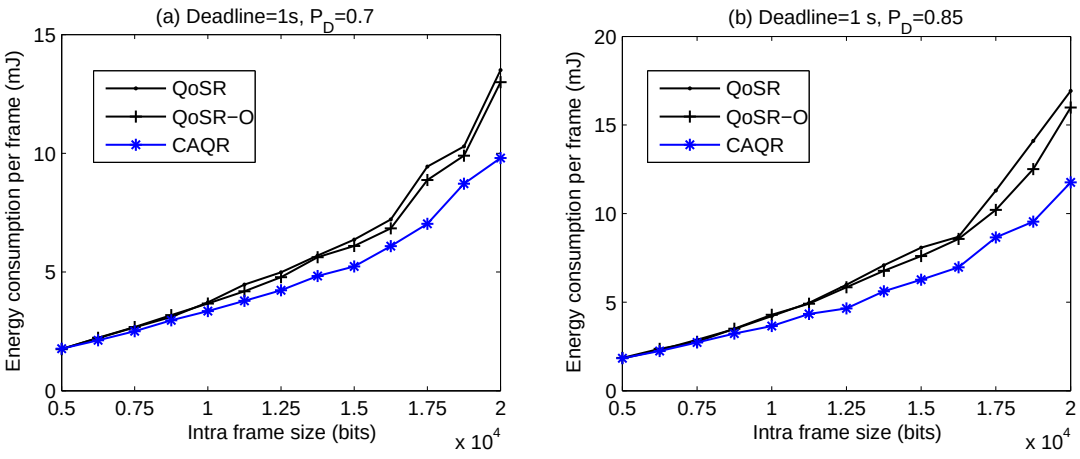

Figure 36: Average energy consumption for correlation-aware QoS routing.

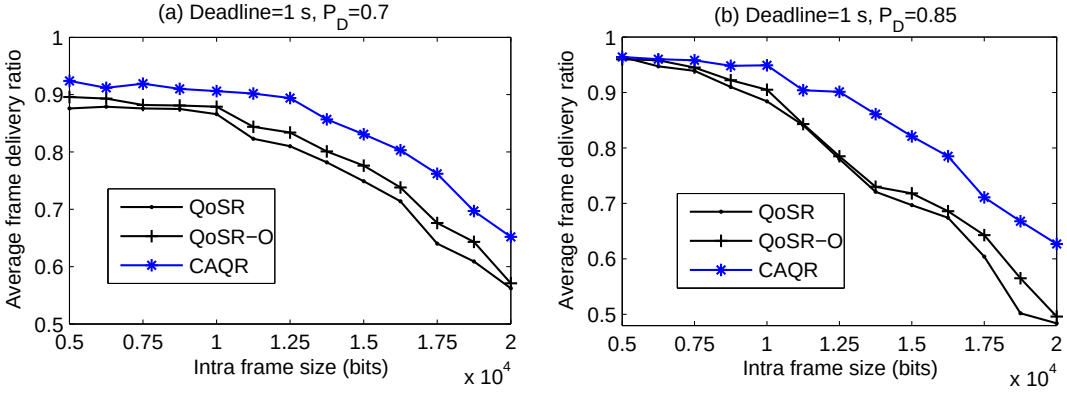

Figure 37: Frame delivery ratio for correlation-aware QoS routing.

routing paths. Fig. 36 shows the average energy consumption for each received image frame. Although differential coding along routing paths introduces extra processing energy, it is performed only when there is considerable energy gain, as indicated in the gain function (64). From the results in Fig. 36, we find that the CAQR and QoSR-O algorithms result in less average energy consumption. The proposed CAQR algorithm is the most energy efficient one since it reduces the transmission of redundant bits as much as possible. We also find that the energy saving for the CAQR algorithm is more obvious when the traffic load is heavy in the network.

We now evaluate the quality of received visual information at the sink under different reliability requirements. We set the deadline to 1 second, and vary the probability

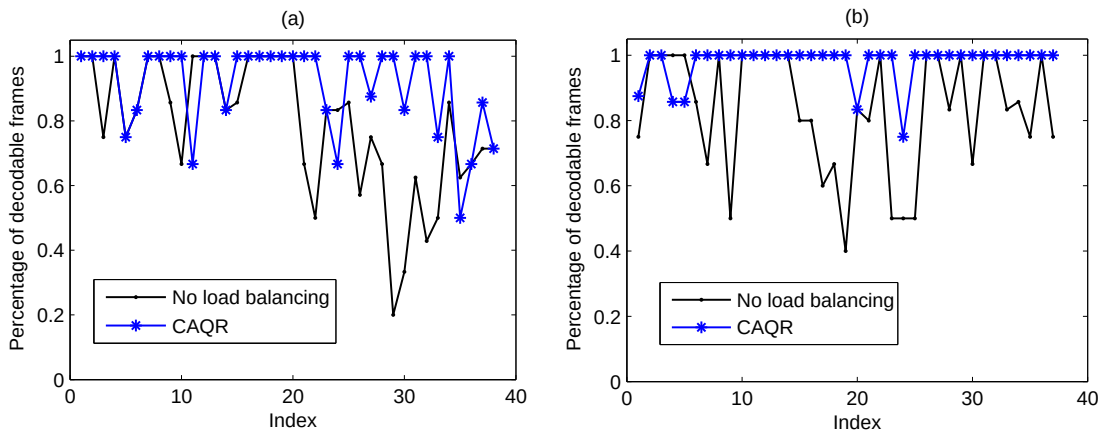

Figure 38: Percentage of decodable frames for different events.

that a video frame is successfully decoded, P_D in Section 5.3.3, to 0.7 and 0.85, respectively. For each reported image frame, we count the number of received packets within the deadline. If the percentage of received packets is above the frame decodable threshold (DT), we deem that this frame is successfully received and decoded at the sink. Based on the number of decoded frames, we can obtain the percentage of successfully decoded video frames (denoted by frame delivery ratio) for each experimental setting. Fig. 37 shows the average frame delivery ratio of the 10 sequences of events under different source coding rates. As the amount of traffic injected into the network increases, the frame delivery ratio decreases. Comparing the results in Fig. 37(a) and Fig. 37(b), the proposed CAQR algorithm shows more advantage when the reliability requirement is high. When the reliability requirement is high, more bits should be transmitted to combat the wireless channel errors, and thus, more benefits can be yielded by reducing the traffic through correlation-aware differential coding. On average, the proposed CAQR algorithm improves the frame decodable ratio by 11.4% compared to QoSR, and by 8.5% compared to the opportunistic QoSR-O algorithm.

In particular, we study the effect of the correlation-aware load balancing operation in Section 5.3.3. We introduce a metric called *event reliability* to facilitate this

evaluation. Suppose at a certain timestamp, multiple correlated video sensors are triggered to report the same event, and each video sensor reports one frame. We denote the total number of frames for this event by N_E. We denote the number of frames that are successfully decoded at the sink by n_E. The *event reliability* is defined as the percentage of decodable frames for an event, denoted by n_E/N_E. We evaluate the event reliability of the whole CAQR algorithm and the CAQR algorithm without correlation-aware load balancing. From the 10 sequences of events mentioned above, we test the performance for two of the sequences that generate relatively larger amounts of traffic. The resulting event reliability for these two sequences of events are plotted in Fig. 38(a) and (b). It can be seen that for several events, the event reliability for CAQR without load balancing has more degradation than the whole CAQR algorithm. This result could be explained from the design of the correlation-aware load balancing scheme. In case of large amount of traffic, a routing path could possibly fail because of congestion. As the correlation-aware load balancing splits correlated images that cannot be further compressed to different paths, multiple images describing the same event could have more chance to be delivered to the sink simultaneously, and in this way the event reliability could be improved.

From the above evaluation on delay, energy, packet delivery ratio, and event reliability, we conclude that exploiting correlation of visual information in a WMSN can enhance network performance, especially when the traffic load is heavy or the QoS requirements are stringent. By incorporating correlation-aware differential coding and load balancing in the routing process, the CAQR algorithm provides an effective way to improve the quality of visual information received at the sink.

CHAPTER VI

CONCLUSION

6.1 Research Contributions

Wireless multimedia sensor networks requires the design of energy-efficient communication protocols while providing quality of service support for various applications. A major challenge to achieve this design goal is the difficulty to process and deliver the huge amount visual information on resource-constrained sensor nodes. To address this challenge, this thesis proposes a correlation-based communication framework by leveraging the spatial correlation of observations from camera sensors in WMSNs.

A novel spatial correlation model for visual information for WMSNs is proposed in Chapter 2. In WMSNs, multiple camera sensors are deployed to provide multiple views, multiple resolutions and enhanced observations of the environment, and there exists correlation among the visual information observed by camera sensors with overlapped field of views. Based on the sensing model and deployment information of camera sensors, a spatial correlation coefficient is derived to describe the degree of correlation between camera sensors, which could be obtained through low computation and communication costs. As part of the correlation model, how much information can be gained from multiple camera sensors is also investigated. The concept of entropy (a measure of the amount of information) is used in the analysis of this problem. We propose a correlation-based approach to estimate the joint entropy of multiple cameras. It is found that cameras with larger degree of correlation lead to less joint entropy. Based on the analysis of joint entropy, a correlation-based camera selection scheme is designed. It selects the minimum number of sensors to report to the sink, such that the amount of information gained at the sink can be maximized.

In Chapter 3, a collaborative image compression framework is designed to reduce the redundancy among correlated cameras. The framework consists of an entropy-based divergence measure (EDM) and a distributed multi-cluster coding protocol (DMCP). The EDM utilizes the results from the above correlation model, and predicts the compression efficiency of performing joint coding of images from correlated cameras. Utilizing the predictions from EDM, the DMCP partitions the entire network into a set of coding clusters, such that the global coding gain is maximized. Simulations show that the framework can bring in promising coding gains. The proposed image compression framework is independent of specific image types and coding algorithms, thereby providing a generic mechanism for image compression in WMSNs. Experimental results on commercial video coding standards have shown that the EDM can effectively predict the efficiency of joint coding, i.e., the percentage of bandwidth that can be saved through coding among correlated cameras. Further simulation results show that using the joint coding efficiency as a clustering metric, the DMCP protocol can achieve better performance than other common clustering algorithms.

In Chapter 4, the correlation of visual information is utilized to design a network scheduling scheme to maximize the lifetime of WMSNs. The scheduling scheme consists of three components including MinMax Degree Hub Location (MDHL), Minimum Sum-entropy Camera Assignment (MSCA), and Maximum Lifetime Scheduling (MLS). First, the MDHL problem finds the optimal locations to place the multimedia processing hubs, which operate on different channels for concurrently collecting images from adjacent cameras, such that the number of channels required for frequency reuse is minimized. With the locations of the hubs determined by the MDHL problem, the objective of the MSCA problem is to assign each camera to a hub in such a way that the global compression gain is maximized by jointly encoding the correlated images gathered by each hub. At last, given a hub and its associated cameras, the

MLS problem targets at designing a schedule for the cameras such that the network lifetime of the cameras is maximized by letting highly correlated cameras perform differential coding on the fly. Approximation and heuristic algorithms are proposed to solve the three problems, and their corresponding performances are evaluated through simulations.

To further exploit the correlation characteristics in WMSNs in communication protocols, in Chapter 5, a correlation-aware QoS routing algorithm is introduced to efficiently deliver visual information under QoS constraints. First, a correlation-aware inter-node differential coding scheme is introduced to remove traffic redundancy along routing paths. Then, a correlation-aware load balancing scheme is proposed to prevent network congestion by splitting the correlated flows that cannot be reduced to different paths. These correlation-aware operations are integrated into an optimization QoS framework that minimizes energy consumption subject to delay and reliability constraints. It has been shown in simulation results that correlation-aware QoS routing improves the energy efficiency and the received quality of visual information at end users.

6.2 Future Work

In the future we intend to broadly address the problem of multimedia communication and processing in WMSNs. An intrinsic feature of wireless multimedia sensor networks is the coexistence of heterogeneous types of traffic such as video streams, snapshot images, audio streams, and scalar data. Different types of sensors capture different aspects of the environment, and they can provide complementary information which is not available from a single type of sensor. Heterogeneous types of traffic may have different quality of service requirements and may subject to different transmission and scheduling policies. As an extension of the work in this dissertation, we can exploit the correlation that exists among different types of media to design

efficient communication protocols. Furthermore, we can perform analytical study on the relationships among multimedia in-network processing, energy consumption, and QoS networking protocols, and as a result of this study, optimal solutions will be developed that integrates multimedia in-network processing together with the underlying networking protocols. These research directions are explained in more details below.

- *Inter-media correlation-based communication.* Apart from the correlation of visual information mentioned above, different types of observations may be correlated with each other, e.g., video and audio signals could be triggered by the same source or collected over a common geographical region. If the observations from different media are correlated, they can be grouped together to improve the performance of multimedia applications. For example, many target detection and tracking systems employ both audio and video sensors, and it has been shown that the fusion of audio and video measurements can improve the tracking performance [76]. Streams from different types of media may also have different importance levels [7]. In good lighting condition, video analysis may be more useful in detecting human than audio analysis; while in a dark environment, information observed by audio sensors could be more useful. We can study the correlation of information observed by different types of sensors. More specifically, we will address the problems of determining which media streams are correlated, estimating the degree of correlation, and evaluating the confidence levels of different media streams. And we will further develop rate-distortion models for heterogeneous multimedia streams. These studies will then be utilized to develop efficient rate control and scheduling schemes for heterogeneous sensors, so that the limited resources in WMSNs could be properly allocated to different types of media to guarantee quality of service.

- *Integration of collaborative multimedia in-network processing and networking protocols.* The huge amount of multimedia traffic poses a major challenge for the communication in multimedia sensor networks. To reduce the amount of traffic injected into a sensor network, various collaborative multimedia in-network processing schemes have been introduced, where sensor nodes could filter out uninterested data independently or coordinate with each other to aggregate correlated data [5]. The performances of multimedia in-network processing and networking protocols are related in several ways. As the processing of multimedia content needs intensive computation, the energy consumption for processing is not negligible. It is worthwhile to investigate how to allocate sensors' limited power to processing and communication. Multimedia in-network processing removes the redundancy of traffic at the expense of introducing extra processing delay, and the reduced traffic could be more prone to channel errors. Thus, multimedia in-network processing influences the performance of QoS communication protocols, which usually aim to provide QoS support in the real-time and reliability domain. We intend to perform comprehensive analysis on the benefits and costs of multimedia in-network processing, and more importantly, on the trade-offs between in-network processing and providing QoS support of networking protocols. Followed by these analytical studies, we expect to find integrated solutions to optimize the performance of multimedia in-network processing together with the underlying networking protocols. In this way, we can better utilize the limited resources in WMSNs to provide satisfactory quality for various multimedia applications.

REFERENCES

[1] GCC, the GNU Compiler Collection. `http://gcc.gnu.org/`.

[2] JVT Reference Software JM 8.5. Available: `http://iphome.hhi.de/suehring/tml/download/old_jm/jm8.5.zip`.

[3] JMVC Reference Software JMVC 2.5. Available: `http://ftp3.itu.int/av-arch/jvt-site/2008_10_Busan/JVT-AC207.zip`.

[4] Abbasi, A. A. and Younis, M., "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, pp. 2826–2841, Jun. 2007.

[5] Akyildiz, I. F., Melodia, T., and Chowdhury, K. R., "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, pp. 921–960, Mar. 2007.

[6] Akyildiz, I. F., Melodia, T., and Chowdury, K. R., "Wireless multimedia sensor networks: Applications and testbeds," *Proceedings of the IEEE (invited paper)*, vol. 96, pp. 1588–1605, Oct. 2008.

[7] Atrey, P., Hossain, M., El-Saddik, A., and Kankanhalli, M., "Multimodal fusion for multimedia analysis: a survey," *Multimedia Systems*, vol. 16, pp. 345–379, 2010.

[8] Bandyopadhyay, S. and Coyle, E. J., "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *IEEE Computer and Communications Conference (INFOCOM)*, Jul. 2003.

[9] Barton-Sweeney, A., Lymberopoulos, D., and Savvides, A., "Sensor localization and camera calibration in distributed camera sensor networks," in *International Conference on Broadband Communications, Networks and Systems*, pp. 1–10, Oct. 2006.

[10] Chen, M., Leung, V. C. M., Mao, S., and Yuan, Y., "Directional geographical routing for real-time video communications in wireless sensor networks," *Computer Communications*, vol. 30, pp. 3368–3383, Nov. 2007.

[11] Chipara, O., He, Z., Xing, G., Chen, Q., Wang, X., Lu, C., Stankovic, J., and Abdelzaher, T., "Real-time power-aware routing in sensor networks," in *14th IEEE International Workshop on Quality of Service (IWQoS)*, pp. 83–92, Jun. 2006.

[12] Cover, T. and Thomas, J., *Elements of Information Theory*. New York: John Wiley and Sons, 1991.

[13] Cristescu, R., Beferull-Lozano, B., and Vetterli, M., "On network correlated data gathering," in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2004*, pp. 2571–2582, March 2004.

[14] Cucchiara, R., "Multimedia surveillance systems," in *Proceedings of ACM International Workshop on Video Surveillance and Sensor Networks*, pp. 3–10, Nov. 2005.

[15] Dai, R. and Akyildiz, I. F., "A spatial correlation model for visual information in wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 11, pp. 1148–1159, Oct. 2009.

[16] Dai, R., Wang, P., and Akyildiz, I. F., "Correlation-aware QoS routing for wireless video sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2010.

[17] Dam, T. and Langendoen, K., "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. of the ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, Nov. 2003.

[18] Devarajan, D., Cheng, Z., and Radke, R., "Calibrating distributed camera networks," *Proc. IEEE*, vol. 96, pp. 1625–1639, Oct. 2008.

[19] Ergen, S. C. and Varaiya, P., "TDMA scheduling algorithms for sensor networks," tech. rep., University of California, Berkley, Jul. 2005.

[20] Felemban, E., Lee, C.-G., and Ekici, E., "MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

[21] Forsyth, D. A. and Ponce, J., *Computer Vision: A Modern Approach*. New Jersey: Prentice Hall, 2002.

[22] Fortet, R., "Applications de lalgebre de boole en recherche operationelle," *Revue Francaise de Recherche Operationelle*, vol. 4, pp. 17–26, 1960.

[23] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, Jan 2005.

[24] Goemans, M. X. and Williamson, D. P., "New 3/4-approximation algorithms for the maximum satisfiability problem," *SIAM Journal On Discrete Mathematics*, vol. 7, no. 4, pp. 656–666, 1994.

[25] Gonzalez, R. C., Woods, R. E., and Eddins, S. L., *Digital Image Processing Using MATLAB*. New Jersey: Prentice Hall, 2004.

[26] Goto, K., Hatabu, A., Nishizuka, H., Matsunaga, K., Nakamura, R., Mochizuki, Y., and Miyazaki, T., "H.264 video encoder implementation on a low-power DSP with low and stable computational complexity," in *IEEE Workshop on Signal Processing Systems Design and Implementation*, pp. 101–106, Oct. 2006.

[27] Gould, R., *Graph Theory*. Benjamin-Cummings, 1988.

[28] Gu, Y., Tian, Y., and Ekici, E., "Real-time multimedia processing in video sensor networks," *Signal Processing: Image Communication Journal (Elsevier)*, vol. 22, pp. 237–251, Mar. 2007.

[29] He, T., Stankovic, J., Lu, C., and Abdelzaher, T., "SPEED: A stateless protocol for real-time communication in sensor networks," in *Proc. of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 46–55, May 2003.

[30] Heinzelman, W., *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 2000.

[31] Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H., "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the Hawaii International Conference on System Sciences*, Jan. 2000.

[32] Horowitz, M., Joch, A., Kossentini, F., and Hallapuro, A., "H.264/AVC baseline profile decoder complexity analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 704–716, Jul. 2003.

[33] Jain, A. K., Murty, M. N., and Flynn, P. J., "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264–323, Sept. 1999.

[34] Kulkarni, P., Ganesan, D., Shenoy, P., and Lu, Q., "SensEye: a multi-tier camera sensor network," in *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005.

[35] Kulkarni, S. and Arumugam, M., "TDMA service for sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 604–609, 2004.

[36] Lin, S. and Jr., D. J. C., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[37] Liu, J., Adler, M., Towsley, D., and Zhang, C., "On optimal communication cost for gathering correlated data through wireless sensor networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 310–321, Sept. 2006.

[38] Liu, Y., Elhanany, I., and Qi;, H., "An energy-efficient QoS-aware media access control protocol for wireless sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference (MASS)*, Nov. 2005.

[39] Lu, Q., Luo, W., Wang, J., and Chen, B., "Low-complexity and energy efficient image compression scheme for wireless sensor networks," *Computer Networks (Elsevier)*, vol. 52, pp. 2594–2603, Sept. 2008.

[40] Luo, H., Liu, Y., and Das, S. K., "Routing correlated data with fusion cost in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1620–1632, Nov. 2006.

[41] Luo, H. and Pottie, G., "A study on combined routing and source coding with explicit side information in sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, Dec. 2005.

[42] Ma, H. and Liu, Y., "Correlation based video processing in video sensor networks," in *IEEE International Conference on Wireless Networks, Communications and Mobile Computing*, pp. 987–992, Jun. 2005.

[43] Manjunath, B. S. and Ma, W. Y., "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 837–842, Aug. 1996.

[44] Melodia, T. and Akyildiz, I. F., "Cross-layer quality of service support for UWB wireless multimedia sensor networks," in *Proc. of IEEE INFOCOM 2008*, pp. 121–125, Apr. 2008.

[45] Melodia, T. and Akyildiz, I. F., "Cross-layer QoS-aware communication for ultra wide band wireless multimedia sensor networks," *IEEE Journal of Selected Areas in Communications*, vol. 28, pp. 653–663, Jun. 2010.

[46] Misra, S., Reisslein, M., and Xue, G., "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 18–40, 2008.

[47] Nath, S., Ke, Y., Gibbons, P., Karp, B., and Seshan, S., "A distributed filtering architecture for multimedia sensors," Technical Report IRP-TR-04-16, Intel Research, Aug. 2004.

[48] Nunes, M., Grilo, A., Macedo, M., and Redol, L. R. A., "Interference-free TDMA slot allocation in wireless sensor networks," in *IEEE Conference on Local Computer Networks*, pp. 239–241, Oct. 2007.

[49] Pattem, S., Krishnamachari, B., and Govindan, R., "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Third International Symposium on Information Processing in Sensor Networks, IPSN 2004*, pp. 28–35, April 2004.

[50] Pluim, J. P. W., Maintz, J. B. A., and Viergever, M. A., "Mutual-information-based registration of medical images: A survey," *IEEE Transactions on Medical Imaging*, vol. 22, pp. 986–1004, Aug. 2003.

[51] Puri, R., Majumdar, A., and Ramchandran, K., "PRISM: A video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing*, vol. 16, pp. 2436–2448, Oct. 2007.

[52] Raghavan, P. and Thompson, C. D., "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, pp. 365–374, 1987.

[53] Rajagopalan, S. and Vazirani, V. V., "Primal-dual RNC approximation algorithms for set cover and covering integer programs," *SIAM Journal on Computing*, vol. 28, no. 2, 1999.

[54] Saxena, N., Roy, A., and Shin, J., "A QoS-based energy-aware MAC protocol for wireless multimedia sensor networks," in *IEEE Vehicular Technology Conference*, pp. 183–187, May 2008.

[55] Seeling, P., Reisslein, M., and Kulapala, B., "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, pp. 58–78, Third Quarter 2004.

[56] Shih, E., Cho, S., N.Ickes, Min, R., Wang, A., and Chandrakasan, A., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. ACM MobiCom '01*, pp. 272–286, Jul. 2001. 2001.

[57] Slepian, D. and Wolf, J., "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471–480, July 1973.

[58] Soro, S. and Heinzelman, W., "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, no. 640386, pp. 1–21, 2009.

[59] Stann, F. and Heidemann, J., "RMST: Reliable data transport in sensor networks," in *Proc. of IEEE Sensor Network Protocols and Applications (SNPA)*, pp. 102–112, Apr. 2003.

[60] Tian, Y. and Ekici, E., "Cross-layer collaborative in-network processing in multi-hop wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 297–310, Mar. 2007.

[61] Vuran, M. C., Akan, O. B., and Akyildiz, I. F., "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Computer Networks (Elsevier)*, vol. 45, pp. 245–259, Jun. 2004.

[62] Wagner, R., Nowak, R., and Baraniuk, R., "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *IEEE International Conference on Image Processing (ICIP)*, pp. 597–600, Sept. 2003.

[63] Wan, C., Campbell, A., and Krishnamurthy, L., "PSFQ: a reliable transport protocol for wireless sensor networks," in *Proc. of ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, Sept. 2002.

[64] Wang, A. and Chandrakasan, A., "Energy-efficient DSPs for wireless sensor networks," *IEEE Signal Processing Magazine*, pp. 68–78, Jul. 2002.

[65] Wang, P., Dai, R., and Akyildiz, I. F., "A spatial correlation-based image compression framework for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 13, pp. 388–401, Apr. 2011.

[66] Wang, P., Dai, R., and Akyildiz, I. F., "Visual correlation-based image gathering for wireless multimedia sensor networks," in *Proc. of IEEE INFOCOM 2011*, Apr. 2011.

[67] Wang, W., Peng, D., Wang, H., Sharif, H., and Chen, H., "Optimal image component transmissions in multirate wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 976–980, Nov. 2007.

[68] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, Apr. 2004.

[69] Wiegand, T., Sullivan, G. J., Bjntegaard, G., and Luthra, A., "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, Jul. 2003.

[70] Wu, H. and Abouzeid, A. A., "Energy efficient distributed image compression in resource-constrained multihop wireless networks," *Computer Communications*, vol. 28, pp. 1658–1668, Sept. 2005.

[71] Wu, M. and Chen, C. W., "Collaborative image coding and transmission over wireless sensor networks," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 70481, 2007.

[72] Xiong, Z., Liveris, A. D., and Cheng, S., "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, pp. 80–94, Sept. 2004.

[73] Ye, W., Heidemann, J., and Estrin, D., "Medium access control with coordinated, adaptive sleeping for wireless sensor networks," *IEEE Transactions on Networking*, vol. 12, pp. 493–506, Jun. 2004.

[74] Younis, O. and Fahm, S., "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366–379, Oct. 2004.

[75] Yu, W., Sahinoglu, Z., and Vetro, A., "Energy efficient JPEG 2000 image transmission over wireless sensor networks," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 2738–2743, Dec. 2004.

[76] Zhou, H., Taj, M., and Cavallaro, A., "Target detection and tracking with heterogeneous sensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 503–513, Aug. 2008.

[77] Ziviani, A., Wolfinger, B. E., J. F. de Rezende, O. M. B. D., and Fdida, S., "Joint adoption of QoS schemes for MPEG streams," *Multimedia Tools and Applications*, vol. 26, pp. 59–80, May 2005.

VITA

Rui Dai was born in Hubei, China, in November 1984. She received her B.S. and M.S. in Electronics and Information Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004 and 2007, respectively. Since August 2007, she has been working as a graduate research assistant at the Broadband Wireless Networking Laboratory (BWN Lab), School of Electrical and Computer Engineering, Georgia Institute of Technology, under the supervision of Professor Ian F. Akyildiz. Her research interests include wireless sensor networks and multimedia communications.